VICTORIA UNIVERSITY OF WELLINGTON
Te Herenga Waka

School of Mathematics and Statistics
Te Kura Matai Tatauranga

PO Box 600
Wellington 6140
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: sms-office@vuw.ac.nz

The Complexity of Visibility and Art Gallery Theorems.

Douglas Renwick

Supervisor: Nick Brettell

February 18, 2022

Submitted in partial fulfilment of the requirements for the
Bachelor of Science with Honours in Mathematics.

Abstract

Simply stated, the Art Gallery Problem (AGP) is the problem of finding the smallest number of guards that can see every point in a simple polygon $\mathcal{P}$. The guards themselves are represented as points in $\mathcal{P}$. We examine recent advances in the complexity of this problem, including the proof that the AGP is $\exists\mathbb{R}$-complete, which means that it is many to one polynomial time reducible to the problem of solving polynomial equations and inequalities using real variables. Currently, the best algorithms for solving such polynomials (and hence $\exists\mathbb{R}$-complete problems) are highly intractable even for small inputs. We also survey some results for smoothed running times of the AGP, and a practical algorithm that uses ideas from smoothed analysis. The end of this survey discusses the chromatic variant of the AGP. To the best of our knowledge, the chromatic AGP's complexity has not yet been studied in the context of the class $\exists\mathbb{R}$. We prove that it is decidable.

Contents

Chapter 1

Introduction

For a closed simple polygon $\mathcal{P}$, a point $q \in \mathcal{P}$ is visible from a point $p \in \mathcal{P}$ if the closed line segment $\overline{pq}$ is contained in $\mathcal{P}$. A finite point set $\mathcal{G}$ is called a guard set of $\mathcal{P}$ if each point in $\mathcal{P}$ is seen by at least one of the elements of $\mathcal{G}$, where each point in $\mathcal{G}$ is called a guard. The question of finding the smallest guard set for a polygon $\mathcal{P}$ was posed in 1973 by Victor Klee [1]. This problem has since developed into the subdiscipline of visibility and art gallery theorems in computational geometry, including two textbooks on art gallery theorems and visibility algorithms [1, 2]. Indeed, Joseph O'Rourke notes in the most recent edition of the handbook on discrete and computational geometry that

"Over 500 papers have been published on aspects of visibility in computational geometry in the last 40 years." [3]

Loosely speaking, discrete and computational geometry concerns itself with finite collections of objects embedded in a geometric space. As we are interested in the art gallery problem, our geometry will be euclidean, and the objects will be points, segments, lines, and polygons.
The AGP was shown to be NP-hard in 1986 [4], but finding NP-membership remained an elusive problem until 2013. New insights into which complexity class it may belong to arose from a result found in what is called the segment graph recognition problem. A graph G with vertex set $\{v_1, v_2, \ldots, v_n\}$ is an intersection graph of segments if there are segments $s_1, \ldots, s_n$ on the plane such that s_i and s_j intersect if and only if $\{v_i, v_j\}$ is an edge of G. The recognition problem can now be stated, as the question: For a given graph G, does there exist a set of segments on the plane such that their intersection relation is correctly described by G?
What is interesting about this question is that McDiarmid and Müller found that some graphs must be represented by segments with endpoints that have coordinates requiring an exponential amount of digits to be described [5]. It was thought that this was not the case, as Jiří Matoušek noted

"Serious people seriously conjectured that the number of digits can be polynomially bounded, but it cannot." [6]

In attempting to prove NP-membership, we would want the number of digits describing such coordinates to have a polynomial bound, because then verifying that a particular certificate is a segment graph or not can be done by just looking at the coordinates. At this point many researchers became doubtful of NP-membership for the segment graph recognition problem, and turned to a new complexity class called $\exists\mathbb{R}$. This class is equivalent, up

to polynomial time reduction, to solving polynomial equations and inequalties using real variables and integer coefficients. The recognition problem for segment graphs, as well as many other problems in computational geometry, have been shown to be $\exists\mathbb{R}$-complete. A survey of these results can be found in Nicholas Bieker's undergraduate thesis [7].
Sticking to visibility theory, we have seen that the recognition problem for visibility graphs is also $\exists\mathbb{R}$-complete [8]. These are graphs where edges between vertices describe visibility between two objects. Not only do they sometimes use rational coordinates where size is not polynomially bounded, but they sometimes require *irrational* coordinates. Similarly, for the AGP, a polygon has been discovered with an optimal guard set that must use a guard with an irrational coordinate (Abrahamsen et al. [9]). This was then generalized to a result that proves there are infinitely many such cases, as in the following theorem

Theorem 3.4.1: Given any real algebraic number α, there exists a polygon $\mathcal{P}$ with corners at rational coordinates such that in any optimal guard set of $\mathcal{P}$ there is a guard with an x-coordinate equal α.

In the same paper, the Art Gallery Problem was shown to be $\exists\mathbb{R}$-complete. Currently, the only known methods for deciding $\exists\mathbb{R}$-complete problems use algebraic methods, which produce intractable running times even for small inputs. This may cause pessimism for anyone wanting an efficient and precise algorithm for the art gallery problem. However, the story does not end here. While a worst case scenario running time has exceptionally bad performance, it may be the case that these polygons are both *rare* to find, and also *rarely* have similar structure.
A recent, and significant (i.e, Gödel prize winning) development in computer science has been the field of smoothed analysis, which is a form of algorithmic analysis that tells us both something about the structure of the input, and its expected running time. The art gallery problem has, according to Dobbins, Holmsen, and Miltzow, been the first $\exists\mathbb{R}$ complete problem to be analyzed using smoothed analysis.

"The significance of our results is that algebraic methods are not needed to solve the Art Gallery Problem in *typical* instances. This is the first time an $\exists\mathbb{R}$-complete problem was analysed by Smoothed Analysis." [10]

With an increasing number of problems in computational geometry being found to be $\exists\mathbb{R}$ complete, it is not difficult to see why smoothed analysis would be interesting to researchers in this field.
We now summarize the content of our survey. In chapter one, we introduce the art gallery problem, as well as some preliminary computational geometry, complexity theory, and algebra. In chapter two, we look at Abrahamsen, Adamaszek, and Miltzow's proof that the Art Gallery Problem is $\exists\mathbb{R}$-complete. This proof is split into three parts. The first part is proving its membership in $\exists\mathbb{R}$. The second part proves that a decision problem called ETR-INV is $\exists\mathbb{R}$-complete. The goal here was to find a system of polynomial equations that are easier to simulate with an optimal guard set for a polygon. The problem ETR-INV provides us with a system of polynomial equations that are bounded by a closed interval and do not have equations of the form $xy = z$. Instead replacing them with $xy = 1$, hence why this class has "INV" in the title. The third part is to prove that the AGP is $\exists\mathbb{R}$-hard by giving a polynomial time reduction from ETR-INV.
Many parts of this proof are highly technical, and tedious to check. In one case software such as Maple is needed to check its validity. One goal in our coverage of this proof has not been to give a complete description. Instead, our aim has been to skip past the large

case checking, and stick to a rigorous treatment of the more interesting parts of the proof. In chapter 3 we look at how to make the AGP 'more tractable', with an investigation of a smoothed analysis of the art gallery problem, as well as efficient, and exact, algorithms that solve the AGP for types of polygons that may be *typical*. We then end our survey by discussing a particular variant called the *chromatic art gallery problem* and prove its decidability by encoding the problem as a formula in ETR.

This survey may be treated as a guide for recent results in the complexity of visibility problems. These papers often require a background in computational geometry, but we have no such requirements here and we aimed to make the results here more accessible to the mathematically inclined student. We used [11], [12], [13], as resources for the computational geometry described in chapter 2. The preliminaries for this paper are the following. Some basic metric space concepts, comfort with many to one polynomial time reduction proofs, and an undergraduate level of graph theory. Complexity theory definitions will be consistent with the ones given by Sipser [14].

Chapter 2

Preliminaries

2.1 Definitions

2.1.1 Graphs

A *graph* $G(V, E)$ consists of a set V of vertices, a set E of edges and an incidence relation between V and E such that each edge is incident with either 1 or 2 vertices. A *loop* is an edge that is incident to exactly one vertex. If two edges are incident to the same pair of vertices, then they are called *parallel* edges. A graph is *simple* if it contains no loops or parallel edges. All graphs considered in this survey will be simple. A *cycle* is a finite sequence of vertices $v_1 \cdots v_n v_1$ such that each consecutive pair of vertices is connected by an edge, and each vertex appears at most once except for v_1, which appears twice. If a graph with n vertices has no cycles, and $n-1$ edges, we call it a *tree*. The *degree* of a vertex is the number of edges that the vertex is incident to. If a vertex has degree *one* then we call it a *leaf*.

2.1.2 Geometry

Given two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ in $\mathbb{R}^2$, we can express any point on the line $\overleftrightarrow{pq}$ as a linear combination of their coordinates, where the coefficients sum to 1:

$$(1-\alpha)p + \alpha q = \big((1-\alpha)p_x + \alpha q_x, (1-\alpha)p_y + \alpha q_y\big)$$

By adding the additional constraint that $0 \leq \alpha \leq 1$, the set of points generated lie on what we call the *closed line segment* $\overline{pq}$. All closed line segments will just be called line segments from now on. The *interior* of the line segment is when we have values $\alpha \in (0,1)$. The *boundary points* of the line segment are the points when $\alpha \in \{0,1\}$. We say that a set is *convex* if for any two points p, q in the set, their line segment $\overline{pq}$ is contained in the set. The containment need not be a proper containment.

A *Jordan curve* is a non self-intersecting closed curve that partitions the plane into an interior and exterior. A *closed simple polygon* $\mathcal{P}$ is a Jordan curve embedded on the plane consisting of finitely many straight line segments and the region they enclose. From now on we will just use the word "polygon" to describe *closed simple polygons*. The endpoints of each straight line segment are called *vertices*, but we may also call them corners when the context requires that they be distinguished from graph vertices. We denote the jordan curve as the *boundary* of the polygon, $\partial\mathcal{P}$. We can see that the boundary of $\mathcal{P}$ can be represented as a cycle for a graph, which means for an n vertex polygon we will have n edges. All polygons in this survey are assumed to be closed simple polygons unless otherwise specified. A *diagonal* of a polygon is a line segment connecting two vertices of $\mathcal{P}$ and lying in the interior of $\mathcal{P}$, not

touching $\partial\mathcal{P}$ except at its endpoints. Two diagonals are *noncrossing* if they share no interior points. A *triangulation* of a polygon $\mathcal{P}$ is a decomposition of $\mathcal{P}$ into triangles by a maximal set of noncrossing diagonals. A vertex of $\mathcal{P}$ is called *reflex* if its interior angle is greater than π, and *convex* if its interior angle is less than or equal to π.

2.1.3 Visibility

A point $q \in \mathcal{P}$ is visible from a point $p \in \mathcal{P}$ if the closed line segment $\overline{pq}$ is contained in $\mathcal{P}$. The visibility region of p, denoted by $V(p)$, is the set of all points visible from $p, V(p) := \{q \in \mathcal{P} \mid q \text{ is visible from } p\}$. A finite point set $\mathcal{G}$ is called a guard set of $\mathcal{P}$ if each point in $\mathcal{P}$ is seen by at least one of the elements of $\mathcal{G}$, i.e. if $\bigcup_{p\in\mathcal{G}} V(p) = \mathcal{P}$. We refer to a member of $\mathcal{G}$ as a guard. We say that a guard set is *optimal* if it is the smallest possible guard set. A polygon $\mathcal{P}$ is a called *rational polygon*, if for some reference point (0,0), each vertex is expressed with rational valued coordinates. A guard is said to be a *rational guard* if both of its coordinates are rational numbers, otherwise it is called an *irrational guard*. The art gallery problem is stated as follows.

For a *rational polygon* $\mathcal{P}$, how many guards are in the optimal guard set of $\mathcal{P}$?

We shall now assume that all polygons contain a diagonal, and that they can be triangulated. Proofs for those assumptions can be found here [12].

2.2 The art gallery theorem

Lemma 2.2.1. *Given a triangulated polygon $\mathcal{T}$, consider the dual graph $\mathcal{T}^*$. Let v^* be the vertex corresponding to the outer face of $\mathcal{T}$. Then $\mathcal{T}^*\backslash v^*$ is a tree.*

Proof. Suppose for contradiction that $\mathcal{T}^*\backslash v^*$ has a cycle C. Then there is a vertex $u \in \mathcal{T}$ where the edges incident to u are exactly the ones passing through the edges of C, which implies u is enclosed by a cycle and so and is not on the boundary of $\mathcal{T}$, a contradiction. □

Definition 2.2.1 (ear)**.** *Three consecutive vertices a, b, c form an ear of a polygon if ac is a diagonal of the polygon. The vertex b is called the ear tip.*

Lemma 2.2.2. *Every triangulated polygon has at least two ears.*

Proof. We use the fact from graph theory that every tree has at least two leaves. We show that every leaf in a tree corresponds to an ear in a polygon. As a leaf f^* is adjacent to exactly one vertex, then the triangle f has only one diagonal. The other two edges of f must both be boundary edges and so adjacent to the diagonal and to each other. This implies the three vertices in f are consecutive on the boundary of the polygon. □

Theorem 2.2.1 (Art Gallery Theorem, Fisk, [15])**.** *For any given polygon, $\lfloor \frac{n}{3} \rfloor$ guards are sufficient to guard the polygon, and are sometimes necessary.*

Proof. Consider a triangulated polygon $\mathcal{T}$ with n vertices. Now let $P(n)$ be the property that any triangulated polygon on $k < n$ vertices is 3-colourable. Delete an eartip of $\mathcal{T}$. Then we have a triangulated polygon on $n-1$ vertices that has a three colouring. Adding the eartip back along with the two edges it is adjacent to gives us back $\mathcal{T}$, and we can colour the *eartip* with a colour not used by its two adjacent vertices. This completes the induction hypothesis and so $\mathcal{T}$ is 3-colourable. We let our guard set be the vertices that used a colour the least amount of times. By the pigeonhole principle, this colour was used at most $\lfloor \frac{n}{3} \rfloor$ times. To see

that $\lfloor \frac{n}{3} \rfloor$ guards are sometimes necessary, we draw a type of polygon called a comb polygon which is shown in figure 2.2. □

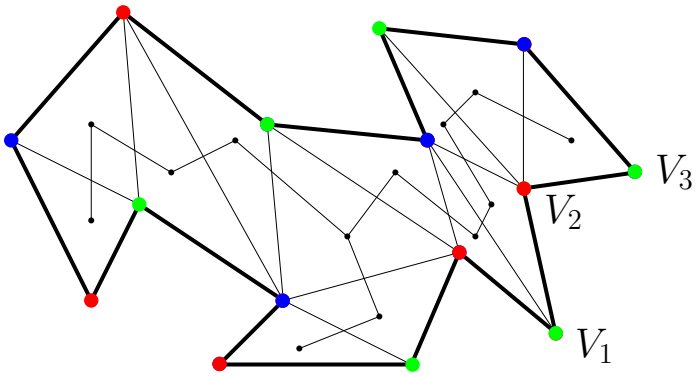


Figure 2.1: A triangulated polygon and its dual after the outerface vertex has been deleted. From the labels we see that V_1 is a convex vertex, V_2 is a reflex vertex, and V_3 is an eartip. Note that the ear contains a leaf.

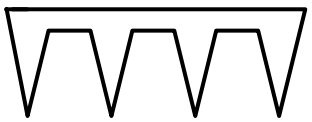


Figure 2.2: A comb polygon, which requires $\lfloor \frac{n}{3} \rfloor$ guards

2.3 Some computational geometry

2.3.1 Point orientation

Recall from linear algebra that the following determinant gives us the cross product of two vectors v_1, v_2 where the vector can be defined as the sum of three orthogonal components $p_i := (x_i, y_i)$. The norm is positive, negative, or zero depending on the order of which vector appears first in the cross product $v_1 \times v_2$. This is the same as calculating the order type of three points on the plane, as shown in Figure 2.3.

$$\det \begin{pmatrix} i & j & k \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}$$

For three points $\langle v_1, v_2, v_3 \rangle$ on the plane, we say that they have *positive orientation* if they define a counterclockwise oriented triangle. If they define a clockwise oriented triangle then they have a *negative orientation*. If they are collinear we get a *zero orientation*. We use the slightly modified determinant below to denote the orientation of an ordered triple. The *sign* just denotes the object we are refering to as having a positive, zero, or negative real value. As our determinants have a fixed size, then they take $O(1)$ time to compute.

$$\text{orient}(v_1, v_2, v_3) = \text{sign} \left(\det \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \right)$$

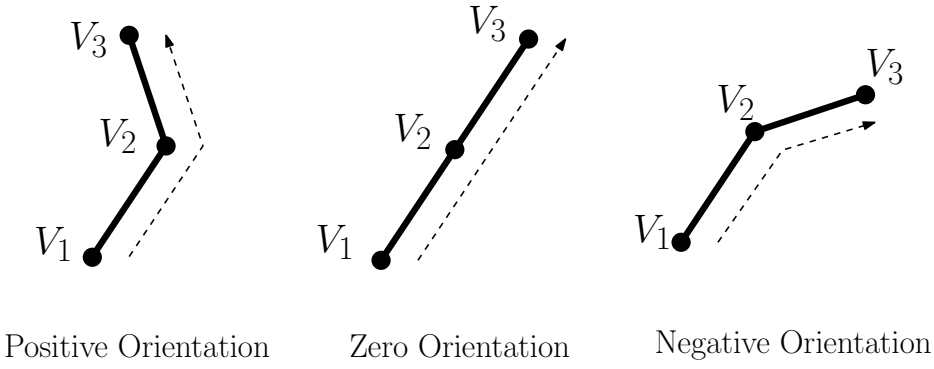


Figure 2.3: Orientated ordered triples

Given we have a polygon $\mathcal{P}$, suppose we want to know which vertices are convex and which are reflex. Let $\langle v_1, v_2, v_3 \rangle$ be a triple of consecutive vertices on the boundary of $\mathcal{P}$. Then v_2 is convex if and only if $\det(\overrightarrow{v_1v_3}, \overrightarrow{v_1v_2}) \geq 0$, and a negative value implies the vertex is reflex.

2.3.2 Point visibility

We say that two segments s_i, s_j *properly intersect* if they intersect at a point that is interior to both segments. Suppose we want to know if a line segment $\overline{v_1v_2}$ is a diagonal of $\mathcal{P}$. One way to do this is to check if $\overline{v_1v_2}$ properly intersects any edge in $\mathcal{P}$. Using this method, we must check that the following two cases are both true (see Figure 2.4, 2.5).

case 1: The line segment $\overline{v_1v_2}$ properly intersects some edge, which we denote as $e = v_iv_j \in \mathcal{P}$.
case 2: The line segment $\overline{v_1v_2}$ is contained inside $\mathcal{P}$.

For case 1, if an edge $e = v_iv_j$ intersects v_1v_2 we will have v_1 and v_2 on opposite sides of e. This is only true if the following two determinants share the same sign:

$\det(\overrightarrow{v_jv_i}, \overrightarrow{v_jv_1})$
$\det(\overrightarrow{v_iv_j}, \overrightarrow{v_iv_2})$

We will also need to have v_i and v_j on opposite sides of v_1v_2 and so the following two determinants must also share the same sign:

$\det(\overrightarrow{v_1v_2}, \overrightarrow{v_1v_i})$
$\det(\overrightarrow{v_2v_1}, \overrightarrow{v_2v_j})$

Now we look at case 2 for a line segment $\overline{v_1v_2}$. To check that $\overline{v_1v_2}$ is inside $\mathcal{P}$, then for the ordered triple $\langle v_l, v_1, v_r \rangle$, let v_l be to the left of v_1, and let v_r be to the right of v_1. Consider the case where v_1 is convex. Then we must have that v_lv_r intersects $\overline{v_1v_2}$. When v_1 is reflex then v_lv_r does not intersect $\overline{v_1v_2}$. By these methods we can conclude that checking if there is a diagonal between two vertices takes $O(n)$ time.

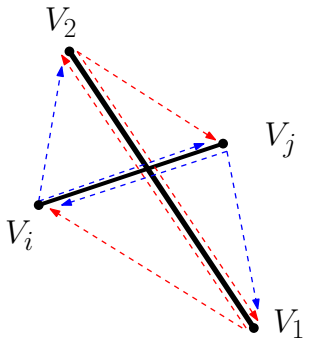


Figure 2.4: A proper intersection of two line segments. The red arrows correspond to two determinants, which must have the same sign. Similarly, this is true for the blue arrows.

2.3.3 Triangulating a polygon

We now have the tools to triangulate a polygon. The following algorithm is called the *ear clipping* algorithm. It works first by finding all the ears in an algorithm, pushing them onto a stack, and then chopping them off the polygon one by one. Once we have found all the ears of a polygon, chopping them off hardly affects the ear status of other triangles.

Lemma 2.3.1. *For a triangulated polygon $\mathcal{T}$, if v_2 is an eartip of v_1, v_2, v_3 then deleting v_2 only changes the eartip status of v_1 and v_3.*

Proof. Let v_j be in $\mathcal{P}\backslash\{v_1, v_2, v_3\}$. Consider the tree $\mathcal{T}^*\backslash v^*$ and let f^* be the vertex in the ear $v_1v_2v_3$. Then deleting v_2 removes the triangle, which is the same as contracting f^* in the tree. It is a fact from graph theory that contracting a leaf in a tree is the same as deleting the vertex and the edge it is incident to. This can only change the leaf status of the vertex f^* was adjacent to, which means v_1 and v_3 are the only vertices that can have their eartip status changed. □

Proposition 2.3.1. *We can locate all ears in a polygon in $O(n^2)$ time.*

Proof. We proceed by consecutively checking each vertex on the boundary of $\mathcal{P}$ if it is convex. There are at most n of them. Every time we encounter a convex vertex, we check if the two vertices adjacent to it give us a diagonal in $\mathcal{P}$, which takes $O(n)$ time. □

This is the bottleneck of our ear clipping algorithm. Once all ears have been pushed onto the stack, then we start to 'clip' the ears off one by one. Updating the ear status takes $O(1)$ time. If we find another eartip, we place a diagonal down and pop the ear from the stack. Then we continue clipping until we have one remaining triangle.

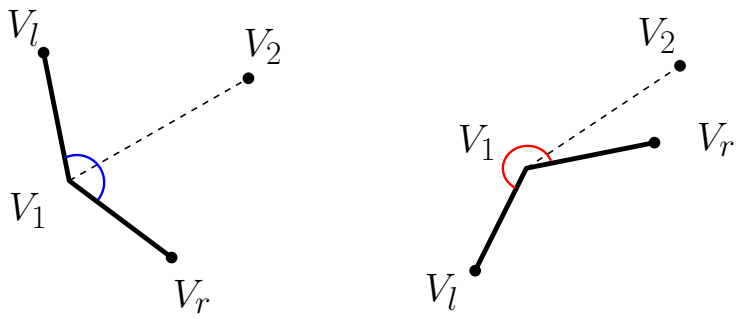


Figure 2.5: The check for case two.

Algorithm 1 Ear Clipping Triangulation Algorithm.

Find all ears in $\mathcal{P}$ and push them on a stack
while $n > 3$ **do**
  locate ear tip v_2
  output diagonal v_1v_3
  pop $v_1v_2v_3$ from the stack
  update ear tip status of v_1 and v_3
end while

2.3.4 Checking if a point is inside a triangle

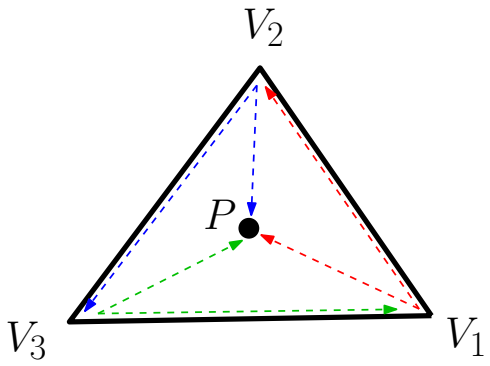


Figure 2.6: Checking if a point is in a triangle. Colour codes correspond to each determinant

Let $\mathcal{T}$ be a triangulated polygon. A point p is in $\mathcal{T}$ if and only if p is in a triangle of $\mathcal{T}$. Consider again the ordered triple $\langle v_1, v_2, v_3 \rangle$ of a triangle T in our polygon. If we want to know whether a point p is inside the triangle, we can evaluate the orientations of three determinants (see Figure 2.6). Each determinant is constructed from an edge of T and the point p. The edges of T are orientated, so that we pick each pair going counterclockwise around the triangle. Then it follows that p is in $\mathcal{T}$ if and only if

$\det(\overrightarrow{v_1v_2}, \overrightarrow{v_1p}) \geq 0$
$\det(\overrightarrow{v_2v_3}, \overrightarrow{v_2p}) \geq 0$
$\det(\overrightarrow{v_3v_1}, \overrightarrow{v_3p}) \geq 0$

2.3.5 Checking if two points see each other in a polygon

To check if two points see each other in $\mathcal{P}$, we use a similar algorithm that checked if a line segment properly intersected an edge, although this time we are checking any arbitrary line segment defined by two points. There are a couple of degenerate cases we must modify this for however. We give such degenerate cases in Figure 2.7. These cases require that we actually take determinants of every two consecutive edges, to check if a line segment grazes the boundary of $\mathcal{P}$ or just goes through a vertex between two edges.

2.4 The existential theory of the reals

Let $X = (X_1, \ldots, X_n)$. Then a formula $\Phi(X)$ in the first order theory of the reals (FTR) uses the symbols from the following alphabet.

$$\{X_1, X_2, \ldots, \forall, \exists, \wedge, \vee, \neg, 0, 1, +, -, \cdot, (,), =, <, \leq\}.$$

From this alphabet the numbers we can construct are limited to the set of integers, and we use $(0, 1, +, -)$ to construct any integer. We can then construct multivariate polynomials with integer coefficents. With the symbols $(=, <, \leq)$ we can compare the polynomials. This amounts to comparing polynomials with zero. Finally, we can connect these comparisons using boolean operators. In summary, this alphabet amounts to a statement involving polynomial inequalities and equations that have integer coefficients. Note that we can easily obtain some extra symbols like $(\geq, >, \Rightarrow)$ as being part of the FTR alphabet by constructing logically equivalent statements from the symbols we are given.
A formula is called a *sentence* if it has no free variables, which means each variable present in the formula is bound by a quantifier. The *first order theory of the reals* is the set of all true sentences described by formulas from the alphabet we described. All sentences using the alphabet of FTR are decidable (Tarski [16]), which means we can use an algorithm to determine their truth in finitely many steps. Each formula Φ in the FTR can be converted to *prenex form*, which means that Φ is quantifier free as all quantifiers are placed in front of the formula.

The existential theory of the reals (ETR) is a set of all true sentences of the first-order theory of the reals in prenex form with existential quantifiers only, i.e., sentences of the form

$$(\exists X_1 \exists X_2 \ldots \exists X_n)\, \Phi\, (X_1, X_2, \ldots, X_n)$$

where Φ is a quantifier-free formula of the first-order theory of the reals with variables $X_1, \ldots, X_n$. The problem ETR is the problem of deciding whether a given existential formula of the above form is true, and is decidable just as FTR is. The complexity class $\exists\mathbb{R}$ consists of all problems that are many to one polynomial time reducible to ETR. Immediately, we can see that the problem ETR is $\exists\mathbb{R}$ complete by definition. To motivate an understanding of $\exists\mathbb{R}$ and ETR, we make the following analogy. The Cook-Levin theorem [17] proved that every problem in NP was polynomial time reducible to SAT. Here we can see there is a similarity between classes $\exists\mathbb{R}$ and NP and the problems ETR and SAT, except for one difference. Unlike for SAT, we don't require an analogous theorem to show ETR is $\exists\mathbb{R}$ complete. It is currently known that

$$\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}.$$

The containment $\exists\mathbb{R} \subseteq$ PSPACE was proven by John Canny in 1988, and we will take it as a given [18]. To see that NP $\subseteq \exists\mathbb{R}$ is much easier, and so we give a proof of this with a polynomial time reduction from the NP-complete class 3SAT.

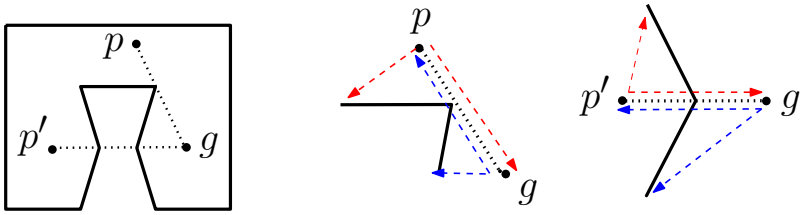


Figure 2.7: g sees p, but not p′. The two figures on the right demonstrate that we can test two appropriate determinants for both consecutive edges in a way that the sign values will distinguish the two cases.

Definition 2.4.1. *(3SAT) Let φ be a boolean formula in 3cnf form, which is when each clause in φ is connected by the $\wedge$ operator and each clause contains three literals connected by $\vee$ operators. Then 3SAT is the decision problem where we must decide if φ is satisfiable, i.e, each clause contains at least one truth assignment.*

Proposition 2.4.1. $NP \subseteq \exists\mathbb{R}$

Proof. The reduction converts a 3 cnf-formula φ into a quantifier free formula Φ with variables $x_1, \ldots, x_n$ so that φ is satisfiable if and only if the problem ETR, with input Φ has output TRUE. Given a set of literals in φ, for any given value of some literal u_i we will define x_i so that $x_i = 1 - u_i$. From this it follows immediately that $x_i = \bar{u}_i$

Now we construct clause gadgets, where the literals and their truth assignments in each clause of φ determine the variables in each clause gadget, so that if $C_a = (u_i \vee \overline{u}_j \vee u_k)$, then our clause gadget is $c_a = (x_i \cdot (1 - x_j) \cdot x_k)$. We also want to force our variables in the formula Φ to have values $x_i \in \{0, 1\}$, and so add the variable gadget $(x_i = 0 \vee x_i = 1)$. Given a sentence φ with m clauses and n variables, our final construction of Φ has m clause gadgets and n variable gadgets, with the following form

$$\Phi = (c_1 + \cdots + c_m = 0) \wedge (x_1 = 0 \vee x_1 = 1) \wedge \cdots \wedge (x_n = 0 \vee x_n = 1).$$

So if φ contains m clauses and n variables, then the number of symbols in Φ is bounded by $O(n + m)$, which means each reduction is done in polynomial time.

Let φ be satisfiable. Then each clause C_i contains some true literal $u_i = 1$ or $\bar{u}_i = 1$, so that each clause gadget c_i contains $x_i = 0$ or $1 - x_i = 0$ respectively. This means each $c_i = 0$ and that $c_1 + \cdots + c_m = 0$. We also see that each $(x_i = 0 \vee x_i = 1)$ if and only if $(x_i = 0)$ or $(1 - x_i = 0)$ if and only if $(u_i = 1 \vee u_i = 0)$ by the way we constructed them. So Φ is true.
Conversely, suppose there exist values for the variables $x_1, \ldots, x_n$ such that Φ is true. Then by the construction of Φ we have $x_i = 0 \vee x_i = 1$ for all i. This means each c_k is multiplied by $1's$ and $0's$, and so for each k we have $c_k = 1$ or $c_k = 0$. But since $c_1 + \cdots + c_m = 0$ this implies each clause gadget c_k evaluates to 0, and so each clause gadget contains a factor x_i or $1 - x_i$ that evaluates to 0, which means the corresponding clause in φ contains $\bar{u}_i = 1$ or $u_i = 1$, meaning each clause is true and so φ is satisfiable. □

2.5 Semi algebraic sets

Define the following set

$$V(\Phi) := \{\mathrm{x} \in \mathbb{R}^n : \Phi(\mathrm{x})\}$$

Then we see that deciding a sentence in ETR is the same as checking if the solution space of $V(\Phi)$ is non empty. A set $S \subset \mathbb{R}^n$ is called *semi-algebraic* if $S = V(\Phi)$. We provide an example for a sentence in ETR and its semi-algebraic set that uses just one real variable. This will mean the solution space will exist on the real line. For example, let $(\exists X)((X^2 - 2 > 0) \vee (X - 1 > 0))$ be a sentence in ETR. Then we have $S = \{(\infty, -\sqrt{2}) \cup (1, \infty)\}$ as our solution space.

Definition 2.5.1. *(connected components-topology) We say that a set X is connected, if it cannot be divided into two disjoint non-empty open sets.*

The following proposition should provide some intuition for the geometric behavior of sentences in ETR, as well as proving an important fact.

Proposition 2.5.1. *A semi algebraic set in $S \subseteq \mathbb{R}^1$ defined by a quantifier free formula of length L has $O(L)$ connected components.*

Proof. We use the fact from linear algebra that a real polynomial of finite degree n has at most n roots. Denote "$\circ$" as one of the operators in $\{\neg \vee, \wedge\}$ and "rel" as one of the operators in $\{>, \geq, 0\}$. As $\Phi = p_1(X)$ rel $0 \circ \ldots \circ p_m(X)$ rel 0, then consider the polynomial among these that has the highest degree, say it has degree n. Then we have at most nm roots for all of the polynomials in Φ. Consider all such roots on the real line. Between any two distinct roots, we could have one or more polynomials specifying that $p(x) > 0$ or $p(x) < 0$. For the relation $p(x) \geq 0$, we can see that this is the union of two connected components, $p(x) > 0 \cup p(x) = 0$. So each connected component is a point or an open interval, where a point has sign 0. For two roots a, b, the interval (a,b) defined by $\circ\ p(x) > 0$ is restriced to a single signed value, either $+$ or $-$. Then S specifies at most nm intervals on the real line where the sign can change for any of the polynomials $p(x)$ rel 0 in S. We saw that any polynomial of degree n uses at least n symbols. So the number of connected components for the formula Φ of length L is $O(L)$. □

Definition 2.5.2. *(description complexity) The description complexity, or complexity, of a sentence Φ in ETR is the number of symbols in Φ, and is denoted $|\Phi| = L$. The complexity of a semi-algebraic set S is the minimal complexity of a formula Φ such that $V(\Phi) = S$.*

The description complexity of formulas in ETR starts to play a significant role in proving polynomial time reductions. To show that a formula Φ is polynomial time reducible to some formula Ψ we generally do so by showing the length of a formula Φ increases in polynomial size. Typically the operations involved in the formula construction will have a low running time.
Lets look at some examples minimizing complexity for sentences in ETR. The number 13 can be expressed using $1 + 1 + \cdots + 1$ easily enough. But $(1 + 1 + 1 + 1) \cdot (1 + 1 + 1) + 1$ uses a smaller number of symbols. More generally, any integer n can be expressed using $O(\log n)$ symbols in our alphabet. Powers are not allowed in our alphabet, so an expression X^k must be constructed from multiplying X by itself k times. For squaring a polynomial, ordinarily we would experience a quadratic increase in complexity if we expressed them in standard form. But we can express the squaring by keeping them in brackets and concatenating one after the other with multiplication. Then their complexity in our formula roughly doubles in size.

Chapter 3

The art gallery problem is $\exists\mathbb{R}$-complete

3.1 The art gallery problem is in $\exists\mathbb{R}$

Theorem 3.1.1. *(Abrahamsen et al. [19]) The art gallery problem is in* $\exists\mathbb{R}$

The Art Gallery Problem was not even proven to be decidable until recently, which was done by encoding it as a formula in the FTR (Efrat, Har-Peled [20]). Unfortunately, converting that formula into one that is in ETR requires constructing a fairly unwieldy formula, and we have to deal with a lot of degenerate cases. Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow said that their goal in constructing this formula was to make it as short as possible [19]. We do not give a complete proof here, and refer to the original paper which has a self contained proof. The construction involves using a witness set that is seen by the set of guards. The witnesses act as points in the polygon which at least one guard must see. We partition the polygon with an arrangement of lines $\mathcal{L} = \{\ell_1, \ldots, \ell_m\}$ where we place a line through each edge of $\mathcal{P}$ and from each guard $g \in \mathcal{G}$ to every corner $v \in \mathcal{P}$. These lines in $\mathcal{L}$ divide the plane into regions, which are connected components of $\mathbb{R}^2 \setminus \bigcup_{\ell \in \mathcal{L}} \ell = \mathcal{A}$. The idea is that we will place at least one witness in each region $R \in \mathcal{P}$, and if at least one guard sees a witness, then that guard will see the entire region. Then we will be able to tell if a guard set sees $\mathcal{P}$ or not. The following proposition ensures that what we just said is true.

Proposition 3.1.1. *Let $\mathcal{P}$ be a polygon, $\mathcal{G}$ be a guard set of $\mathcal{P}$ and $\mathcal{A}$ be the set of regions as defined above. Then the following are true.*

1. Each region in $\mathcal{A}$ is an open convex polygon.
2. Each region in $\mathcal{A}$ is either contained in $\mathcal{P}$ or contained in the complement of $\mathcal{P}$
3. The closure of the union of the regions that are contained in $\mathcal{P}$ equals $\mathcal{P}$.
4. For each region $R \in \mathcal{A}$, each guard $g \in \mathcal{G}$ either sees all points of R or sees no point in R. In particular, if a guard g sees one point in R, it sees all of R and its closure.

Proof. Let q be a point in a region R. Then $q \notin \ell$ for any line $\ell \in \mathcal{L}$. Take the orthogonal projection of q to all lines in $\mathcal{L}$. The orthogonal projection is the closest point on each line to q. Consider the distance from q to each orthogonal projection and set ε to be the smallest such distance. Then $B(q, \varepsilon) \subseteq R$ and so R is open.
Now suppose for contradiction that R is not convex. Then R contains a reflex vertex. But this is not possible, as each corner is the intersection of two lines that are infinite in each direction, which would mean two lines intersect the interior of R, giving us the contradiction.

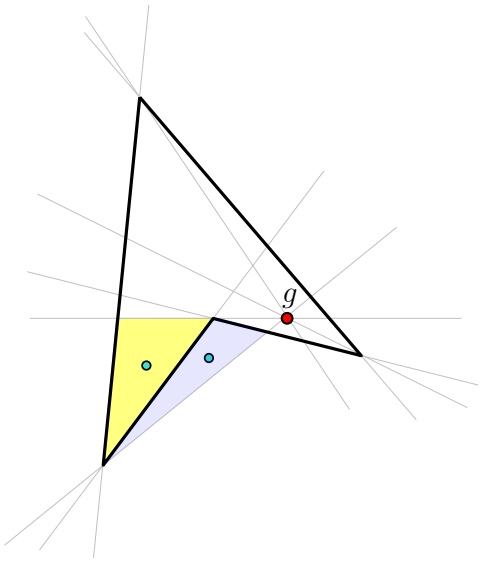


Figure 3.1: An arrangement of lines partitioning our polygon into regions, where g is a guard. The two blue points are examples of centroids defined by the corners of the coloured triangles. The yellow region is not seen by g, and the blue region isn't in $\mathcal{P}$

So R is convex.

Suppose R is a region not fully contained in $\mathcal{P}$ or $\mathbb{R}^2 \backslash \mathcal{P}$. Then R contains a point $p \in \mathcal{P}$ and $q \in \mathbb{R}^2 \backslash \mathcal{P}$. But this implies an edge properly intersects the region, and so a line ℓ does too, which is a contradiction. So $R \subset \mathcal{P}$ or $R \subset \mathbb{R}^2 \backslash \mathcal{P}$.

Define $\mathcal{R} = \overline{\{\bigcup R : R \subseteq \mathcal{P}\}}$, which is the closure of the union of the regions contained in $\mathcal{P}$. By definition $\mathcal{R} = (\bigcup R) \cup \partial(\bigcup R)$. It is a fact from elementary analysis that $\mathcal{R}$ is the smallest set that contains $\bigcup R$. As $\bigcup R \subseteq \mathcal{P}$, then $\mathcal{R} \subseteq \mathcal{P}$.
Now let $p \in \mathcal{P}$. Then either $p \in R_j \subseteq \bigcup R \subseteq \mathcal{R}$ for some region $R_j \in \bigcup R$ or $p \in \ell_i \cap \mathcal{P}$ for some line $\ell_i \in \mathcal{L}$. As we want to show that $p \in \mathcal{R}$, we need only consider the later case. Let $\varepsilon > 0$ and consider the ball $B(p, \varepsilon)$ where $p \in \ell_i \cap \mathcal{P}$. The line ℓ_i forms a chord on the ball, and from this fact we can see that the chord is incident to at least two regions of $\mathcal{A}$, where one of the regions must be in $\mathcal{P}$. Let q be a point where $q \in B(p, \varepsilon) \cap R_i$ where R_i is such a region contained in $\mathcal{P}$. Then $q \in \bigcup R$ and $p \notin \bigcup R$. This implies $p \in \partial(\bigcup R)$, which means $p \in \mathcal{R}$. So $\mathcal{P} \subseteq \mathcal{R}$ and both containments give us $\mathcal{R} = \mathcal{P}$.

Let p, q be points in a region R and $g \in \mathcal{G}$. Suppose g sees p and g doesn't see q. Then there is an edge $e \in \mathcal{P}$ properly intersecting $\overline{gq}$ but not $\overline{gp}$. As e has a line passing through it, then that line creates two halfplanes, and p, q must both be in one of those halfplanes, otherwise they are in different regions, which is a contradiction. So p, q are on the same side of the line passing through e while g is on the other side. As g sees p, then e must have a vertex v inside the triangle gpq. This means that a line passes through g and v, implying that p and q are on separate half planes created by the line, again we get the same contradiction as before. So g must see q, and as we chose q arbitrarily, this implies g sees all points in R if and only if g sees a single point in R. □

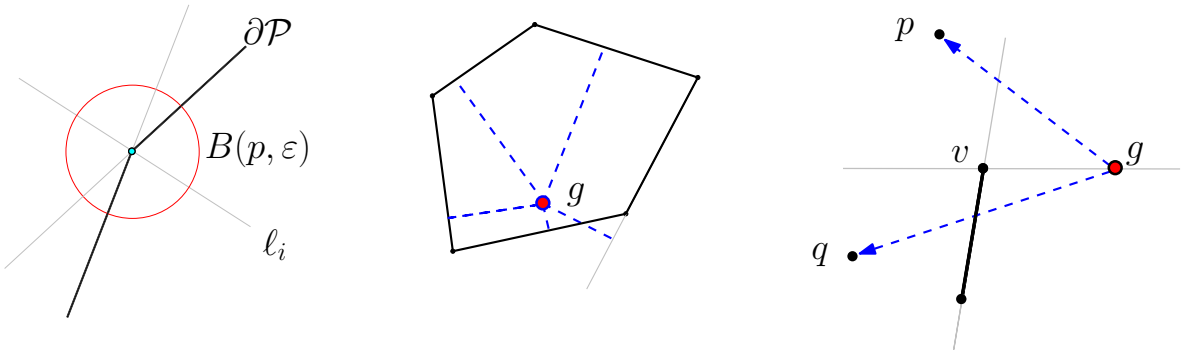


Figure 3.2: On the left we have the point $p \in \ell_i \cap \mathcal{P}$. At the centre figure we have blue dotted lines intersecting orthogonal projections onto each line. On the right is the contradiction we see when the line passing through $\overline{gv}$ intersects the region p and q are in.

3.1.1 A partial construction of the formula Φ

Let $\mathcal{G}$ be a guard set of size k. For each $i \in \{1, \ldots, k\}$, the variables x_i, y_i represent the position of guard $g_i := (x_i, y_i)$, and $p := (p_x, p_y)$ represents an arbitrary point. The predicate INSIDE-POLYGON(p_x, p_y) tests if the point p is contained in the polygon $\mathcal{P}$, and SEES(x_i, y_i, p_x, p_y) checks if the guard g_i can see the point p. In the last section we saw that a we could triangulate a polygon to check if a point was inside $\mathcal{P}$. We also saw how to decide if the line segment $\overline{pg_i}$ was inside a polygon or not. The formula Ψ was used to encode the AGP as an instance of the FTR. We now convert it into a formula in ETR by replacing the $\forall$ quantifiers with a witness set. By our arguments from computational geometry, we can see that Ψ is satisifiable if and only if our guard set of size k sees every point $p \in \mathcal{P}$.

$$\Psi := \left[\exists x_1, y_1, \ldots, x_k, y_k \forall p_x, p_y : \text{INSIDE-POLYGON}\left(p_x, p_y\right) \implies \bigvee_{i=1}^{k} \text{SEES}\left(x_i, y_i, p_x, p_y\right)\right]$$

Let $(i, j) = \{1, \ldots, q\}^2$ be a pair that represents two lines in $\mathcal{A}$. A line is said to be well defined if it contains two distinct points on the plane. For each well defined and non parallel pair of lines, we define their intersection point as $X^{ij} := (x_{ij}, y_{ij})$. For three intersection points X^{ab}, X^{cd}, X^{ef} we define the centroid as $(X^{ab} + X^{cd} + X^{ef})/3$. Let X be the set of all such centroids that use well defined and parallel lines. Observe that every region in $\mathcal{P}$ must contain at least one centroid. So X will be our witness set, i.e, the set of points that need to be seen by at least one guard. We obtain the following bound on the number of centroids.

Proposition 3.1.2. *There are $O((kn)^6)$ centroids of three points in X.*

Proof. Let there be n corners and k guards in $\mathcal{P}$. Then there are n lines from each guard to each corner, so for k guards we get nk lines. We also have a line passing through each edge, so there are an additional n lines. In total we have $nk + n$ lines. Then $nk + n$ lines intersect $(nk + n)^2$ at most times. The actual number is less than this, but we will encode for double ups as well. So we have a bound of $O((kn)^2)$ on the number of intersection points.
For each triple of intersections we can define at most one centroid of a triangle, which implies we get this bound on the number of centroids.

$$\binom{O((kn)^2)}{3} = O(((kn)^2)^3)$$

□

The formula described in the next section will have predicates for each centroid, as well as a predicate checking to see if each centroid is seen. Then $|\Phi|$=L=O($(kn)^7$)

3.1.2 An informal description of Φ

It is not our desire to fully describe Φ. To give a complete formula that correctly encodes the Art Gallery Problem we need to deal with some degeneracies. This in turn makes the formula fairly unwieldy, and in turn a lot of truth assignments are needed to be checked to see that it works. We refer to [19] for the full proof and formula. The whole formula Φ is used in Proposition 4.3.2.
We note some important predicates used in Φ, however. The predicate INTERSECT(i, j) checks that the variables x_{ij}, y_{ij} are the correct intersection coordinates for a well defined and non parallel pair of lines (ℓ_i, ℓ_j). Another two predicates are used to check that centroids are well defined and have the right coordinates.

3.2 Reductions from ETR to ETR-INV

The polynomials of a formula in ETR give a solution space that is very dfficult to simulate with a solution to the art gallery problem. The key step in the $\exists\mathbb{R}$ completeness proof is to bound the variables of a formula to a closed space where every element has an inverse. In this case they pick the space $[1/2, 2]^n$ for a formula on n variables. So each variable is represented on the interval $[1/2, 2]$. Then multiplication of variables $xy = z$ in a polynomial is replaced by $xy = 1$ to simulate inversion instead.

Definition 3.2.1. *($ETR^{c,+,\cdot}$)*
$ETR^{c,+,\cdot}$ is the problem where $c \in \mathbb{R}$, we are given a set of real variables $\{x_1, \ldots, x_n\}$, and a set of k equations, where each equation has one of the following forms:

$$x = c, \quad x + y = z, \quad x \cdot y = z,$$

for $x, y, z \in \{x_1, \ldots, x_n\}$. The goal is to decide whether the system of equations has a solution. A modified version of the problem, where we additionally require that $x_1, \ldots, x_n \in [a, b]$ for some $a, b \in \mathbb{R}$, is denoted by $\mathrm{ETR}^{c,+,\cdot}_{[a,b]}$.

Definition 3.2.2. *(ETR-INV)*
ETR-INV is the problem where we are given a set of real variables $\{x_1, \ldots, x_n\}$, and a set of k equations, where each equation has one of the following forms:

$$x = 1, \quad x + y = z, \quad x \cdot y = 1,$$

for $x, y, z \in \{x_1, \ldots, x_n\}$. The goal is to decide whether the system of equations has a solution when each variable is restricted to the range $[1/2, 2]$.

To show that ETR-INV is $\exists\mathbb{R}$ complete, we take a formula Φ in ETR and give a sequence of polynomial time reductions where each reduction slightly modifies the previous formula, until we arrive at ETR-INV. To begin with, first we reduce Φ to a single polynomial equation. Then we take the polynomial equation and replace all variables to get a formula in $\mathrm{ETR}^{1,+,\cdot}$. Afterwards we take a formula and shrink all of the variables down so they are close to the origin in $\mathbb{R}^2$. Once this is acheived we shift those variables onto the interval $[1/2, 2]$ to get a formula in $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$. Finally, we replace multiplication with squaring and addition, and then replace squaring with inversion and addition. This gives us a formula in

ETR-INV as desired. We provide a diagram that gives an overview of this proof in Figure 3.3.

Definition 3.2.3. *ETR-FEASIBLE is the problem where we must decide if there is a solution to a single polynomial equation that uses integer coefficients and is defined using a set of real variables* $\{x_1, \ldots, x_n\}$.

Abrahamsen et al. [21] have since provided a toolbox that gives a more generalized and self-contained description of the reductions that are used here, which a researcher may find helpful. We used some techniques from that paper to give a somewhat different proof of the first reduction in our sequence. Other proofs of feasisbility in the literature can be found here [6], [22].

Proposition 3.2.1. *Given an ETR formula* Φ *of complexity L, we can produce an* $\mathrm{O}(\mathrm{L})$ *length formula* $\Psi = p(Z) = 0$ *with* $\deg(p(Z)) \geq 4$

Proof. We do so by replacing each subformula which uses an operator that isn't used to describe polynomials, with an equivalent subformula using operators that describe polynomials. To do this, we will do the following in a strict order. (1) replace $\neg$, (2) replace $>$, (3) replace $\geq$, (4) replace $\vee$, and (5) replace $\wedge$. We use p, q to denote polynomials in our proof.

To replace $\neg$, we push negation down the formula until it reaches the atomic predicates of Φ. Each such negated predicate can be replaced with an equivalently non negated one as follows:

$$\begin{array}{lcl} \neg(q > 0) & \mapsto & -q \geq 0 \\ \neg(q = 0) & \mapsto & (q > 0) \vee (-q > 0) \\ \neg(q \geq 0) & \mapsto & -q > 0 \end{array}$$

To replace $p > 0$, we use the fact that for any real number, there exists a multiplicative inverse of that number. Then we introduce two variables to get

$$(\exists z)(\exists z')(p \cdot z - 1 = 0 \wedge z'^2 - z = 0)$$

To replace $p \geq 0$ we introduce two variables to get

$$(\exists z)(\exists z')(p - z = 0 \wedge z'^2 - z = 0)$$

Suppose we encounter $\Phi_1 \vee \Phi_2$ for two subformula's Φ_1, Φ_2 contained in Φ. As we have already eliminated the symbols $\{\neg, >, \geq\}$, then either of these subformula's will contain one or more polynomials connected by the boolean operators $\{\wedge, \vee\}$. Let $i \in \{0, 1\}$. We consider each case:

(i) Suppose Φ_i contains $p = 0 \wedge q = 0$. Then replace with

$$p \cdot p + q \cdot q = 0$$

.
(ii) Suppose Φ_i contains $p \vee q = 0$. Then replace with

$$p \cdot q = 0$$

We repeat this step until Φ_i represents a single polynomial equation $p = 0$.

So far our replacements have produced a formula of the following form:

$$p_1 = 0 \wedge \cdots \wedge p_m = 0 \wedge z_1'^2 - z_1 = 0 \wedge \cdots \wedge z_k'^2 - z_k = 0$$

where p_i is one of the polynomials we introduced in steps (1)-(4). We are left with a formula that is a conjunction of polynomial equations. From here we eliminate the $\wedge$ operator just as we did before to get

$$p_1^2 + \ldots + p_m^2 + (z_1'^2 - z_1)^2 + \ldots + (z_k'^2 - z_k)^2 = 0$$

For $|\Phi| = L$, we can have no more than L subformulas and predicates. For each replacement, we introduced $O(1)$ symbols. Therefore Φ reduces to a formula Ψ with $O(L)$ length. □

Proposition 3.2.2. *Given an ETR-FEASIBLE formula Φ of complexity L, we can produce an $O(L)$ length formula Ψ in* $\mathrm{ETR}^{1,+,\cdot}$.

Proof. Let $\Phi = p(X) = 0$, where $X = (x_1, \ldots, x_n)$. Our aim will be to turn $p(X)$ into some polynomial $q(Z) = 1$ using variables $Z = (V_1, \ldots, V_m, z_1, \ldots, z_k)$. We need to construct each integer coefficient of $p(X)$ using equations of the form

$$x = 1, \quad x + y = z, \quad x \cdot y = z.$$

Start by defining $V_1 = 1$. For any integer $N \in \mathbb{N}$, we can represent V_N using variables $V_n + V_m$ or $V_n \cdot V_m$ where n, m are integers. For example if we want a variable representing 5, we can let $2 = V_2 = V_1 + V_1$, and $4 = V_{2^2} = V_2 \cdot V_2$, then finally $5 = V_{2^2} + V_1$. When we have two variables of the form $x_j \cdot x_k$ or $x_j + x_k$ in $p(X)$, we can iteratively replace these variables with some z_i until we have a single variable left, turning $p(X) = 0$ into $q(Z)' = z_k = 0$. Then add V_1 to finally get $q(Z) = z_k + V_1 = 1$. This gives us the reduction to $\mathrm{ETR}^{1,+,\cdot}$. □

The following lemma is a very powerful tool which essentially says that in order push the bounds on the solution space of a semi-algebraic set away from the origin in $\mathbb{R}^n$, we must use more symbols in our formula Φ to do so. We will take the lemma as a given. The radius of the ball described is a simplification established by Schaefer and Štefankovič [22] in order to state the theorem in a more readible manner.

Lemma 3.2.1. *(Basu, Roy [23]). Let Φ be an instance of ETR of complexity $L \geq 4$ such that $V(\Phi)$ is a non-empty semi algebraic subset of $\mathbb{R}^n$. Let B be the ball in $\mathbb{R}^n$ at distance at most $2^{L^{8n}} = 2^{2^{8n \log L}}$ from the origin. Then $B \cap V(\Phi) \neq \emptyset$.*

Proposition 3.2.3. *Given an $ETR^{1,+,\cdot}$ formula Φ of complexity L, we can produce an $O(L)$ length formula Ψ in* $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$.

Proof. Let Φ be a formula of $\mathrm{ETR}^{1,+,\cdot}$ with n variables $x_1, \ldots, x_n$ and complexity L. We construct a formula Ψ of $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ such that Φ has a solution if and only if Ψ has a solution. For our intentions, the size of the radius does not matter. So long as one exists. So for yet more simplification, we fix a radius for the ball $B(0, k)$ such that

$$k \geq 2^{2^{8n \cdot \log L}}$$
$$k = 8^m \text{ for some m } \in \mathbb{N}$$

and we fix $\varepsilon := 1/k$. For Ψ, we first define a variable V_ε satisfying $V_\varepsilon = \varepsilon$. Then we define m new variables $V_{1/8}, V_{1/8^2}, \ldots, V_{1/8^m}$ and equations

$$\begin{aligned} V_{1/8} &= 1/8 \\ V_{1/8} \cdot V_{1/8} &= V_{1/8^2} \\ V_{1/8} \cdot V_{1/8^2} &= V_{1/8^3} \\ &\vdots \\ V_{1/8} \cdot V_{1/8^{m-1}} &= V_\varepsilon \end{aligned}$$

In Ψ, we use the variables $V_{\varepsilon x_1}, \ldots, V_{\varepsilon x_n}$ instead of $x_1, \ldots, x_n$ to scale them down to the interval $[-1/8, 1/8]$. An equation of Φ of the form $x = 1$ is transformed to the equation $V_{\varepsilon x} = V_\varepsilon$ of Ψ. An equation of Φ of the form $x + y = z$ is transformed to the equation $V_{\varepsilon x} + V_{\varepsilon y} = V_{\varepsilon z}$ of Ψ. An equation of Φ of the form $x \cdot y = z$ is transformed to the following equations of Ψ, where $V_{\varepsilon^2 z}$ is a new variable satisfying:

$$\begin{aligned} V_{\varepsilon x} \cdot V_{\varepsilon y} &= V_{\varepsilon^2 z}, \\ V_\varepsilon \cdot V_{\varepsilon z} &= V_{\varepsilon^2 z}. \end{aligned}$$

Assume that Φ is true. Then there exists an assignment of values to the vector $(x_1, \ldots, x_n)$ of Φ that satisfies all the equations and $(x_1, \ldots, x_n) \in [0, k]^n$. Then the assignment $V_{\varepsilon x_i} = \varepsilon x_i$ yields a solution to Ψ with $(\varepsilon x_1, \ldots, \varepsilon x_n) \in [-1/8, 1/8]^n$. Conversely, if there is a solution to Ψ, an analogous argument yields a corresponding solution to Φ. We have given a reduction from $\mathrm{ETR}^{1,+,\cdot}$ to $\mathrm{ETR}^{1/8,+}_{[-1/8,1/8]}$. The length of the formula essentially doesn't change aside from m introduced variables, and so increases by at most $\mathrm{O(L)}$. □

The proofs of the next three reductions will be skipped over, as they require a lot of work to check. We will just describe the key ideas.

Proposition 3.2.4. *Given an* $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ *formula of complexity L, we can produce an* $\mathrm{O(L)}$ *length formula* Ψ *in ETR* $^{1,+,\cdot}_{[1/2,2]}$.

proof idea:
This reduction involves shifting all of the variables from the bound $[-1/8, 1/8]$ to $[1/2, 2]$. For each variable $x \in [-1/8, 1/8]$, we introduce a variable $V_{x+\frac{7}{8}} = x + \frac{7}{8}$. This shifts all variables into the appropriate interval and $x = \frac{1}{8}$ becomes $V_{x+\frac{7}{8}} = 1$.

Proposition 3.2.5. *Given an* $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$ *formula* Φ *of complexity L, we can produce an* $O(L)$ *length formula* Ψ *in ETR-INV.*

proof idea:
In this reduction, each equation $x \cdot y = z$ must be replaced by $x \cdot y = 1$. Observe that for that $x \notin \{0, 1\}$ we have $\frac{1}{x-1} - \frac{1}{x} = \frac{1}{x^2-x}$. Therefore, a variable V_{x^2} satisfying $V_{x^2} = x^2$ can be constructed from x using only a sequence of additions and inversions. Similarly, as $(x + y)^2 - x^2 - y^2 = 2xy$, a variable V_{xy} satisfying $V_{xy} = xy$ can be constructed from x and y using a sequence of additions and squarings.

Theorem 3.2.1. *(Abrahamsen et al. [19]). ETR-INV is* $\exists\mathbb{R}$ *complete.*

All reductions only increased the length of the formula by a factor of $O(L)$, and so we have shown to how to obtain a formula Φ of complexity L in ETR to a formula Ψ in ETR-INV with $O(L)$ complexity. So ETR-INV is $\exists\mathbb{R}$-complete.

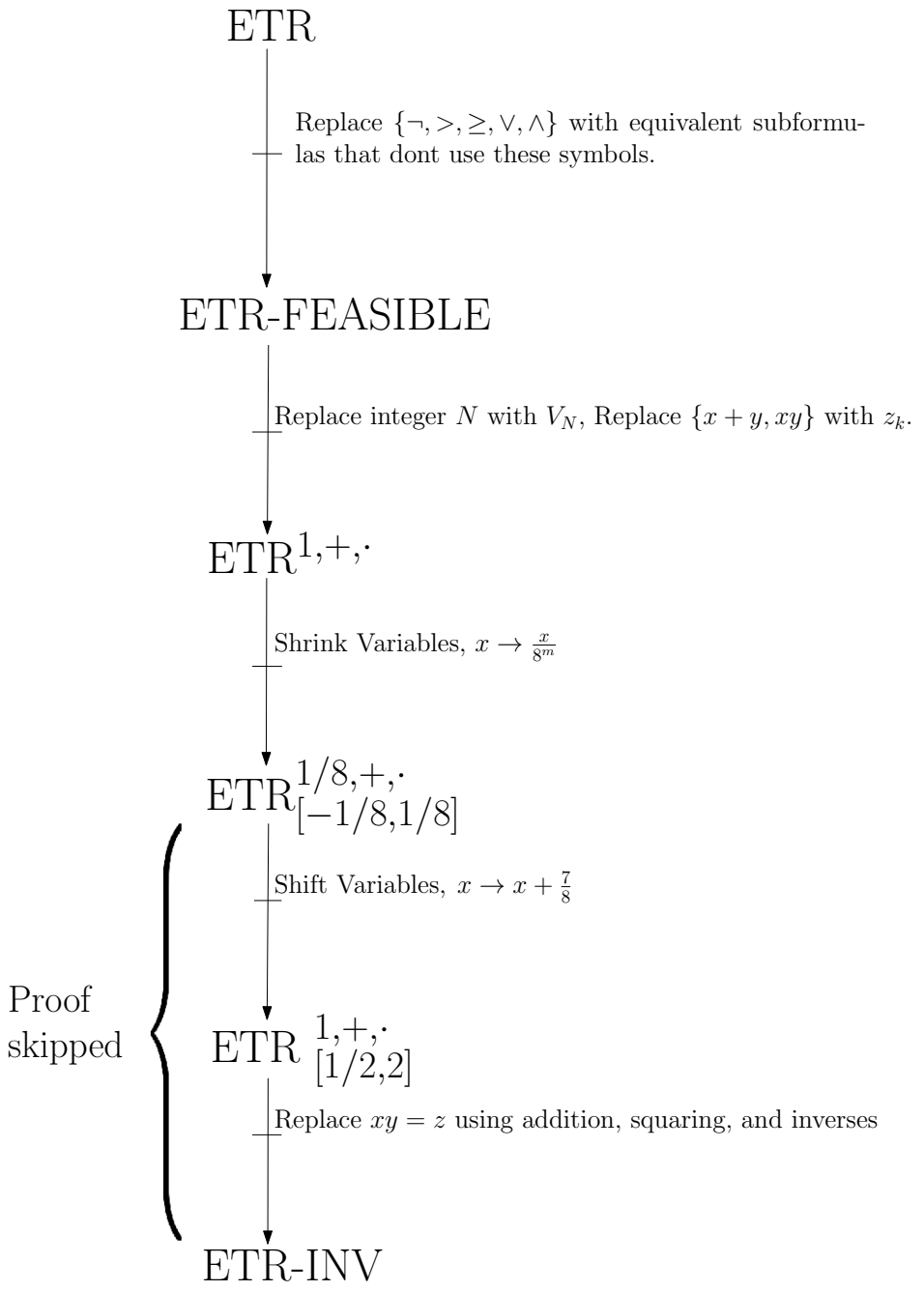


Figure 3.3: Overview of our sequence of reductions

3.3 The art gallery problem is $\exists\mathbb{R}$-hard

3.3.1 An NP-hard proof of the art gallery problem

We cannot cover the entire proof of $\exists\mathbb{R}$-hardness of the art gallery problem, and proving NP-hardness is a lot easier. So as a warm up, we will give a complete proof of NP-hardness. This result was first proved in 1986 by Lee and Lin for the case where guard location was restricted to the vertices in $\mathcal{P}$. They gave a polynomial time reduction from the NP-complete class 3SAT. A slight modification of their proof by Aggarwal made it work for point guards as well [4]. A slightly cleaner version of their proof can be found in O'Rourke's textbook [1]. We provide an alternative proof by giving a polynomial time reduction from the NP complete problem MIN VERTEX COVER.

Definition 3.3.1. *(MIN VERTEX COVER)*
The minimum vertex cover of a graph is the smallest set of vertices such that every edge in the graph is incident to a vertex from the set.

Theorem 3.3.1. *The point guard version of the art gallery problem for closed simple polygons is NP-hard*

Construction of the polygon

Given the input $G(V, E)$ where G is a simple graph and $|V| = n$, we construct a polygon $\mathcal{P}$ where the boundary has the structure of a $2n$ sided regular polygon. The polygon is shown in Figure 3.5. The midpoint between every two consective sides of the boundary corresponds to the vertices in G, and we will be placing the point guards that correspond to vertices from the cover near the location of these midpoints. For each edge in G we modify the frame by cutting two rectangular wells into the boundary around the line that passes between the two midpoints. This represents the vertices that the edge is incident to. So far this description is enough to give us the complete incidence relation of the vertices and edges for the graph G.
Near the top of $\mathcal{P}$ we also modify the boundary so that we have a well for each vertex in G, with the goal being to create n pairwise disjoint regions. This implies we will need at least n point guards. In fact our goal will be to ensure that n point guards are sufficient. The next step is to split the point guards off into two types of regions, where each of the two regions correspond to point guards used in a min vertex cover and ones which are not. We will describe four important convex regions that arise from how we have modified the boundary of $\mathcal{P}$, and to see the union of these regions will mean that we see $\mathcal{P}$.

Construction of the regions

We have n vertices in G, and n corresponding critical points located at every two consecutive edges of the regular polygon framework with $2n$ sides. They are labelled by starting at the top right of the polygon and going around clockwise to get $v_1, \ldots, v_n$. If $v_i v_j$ is an edge in G, then we take the line ℓ_{ij} which passes through critical points v_i, v_j. Where ℓ_{ij} intersects the boundary, we modify the boundary to get a rectangular shape which we informally describe as a "well", and ℓ_{ij} bisjects this well. The size of the rectangle is as small as it needs to be to avoid two overlapping rectangles in the construction. The *edge region* is the set of points that can be seen by all points in the well, and so is convex.

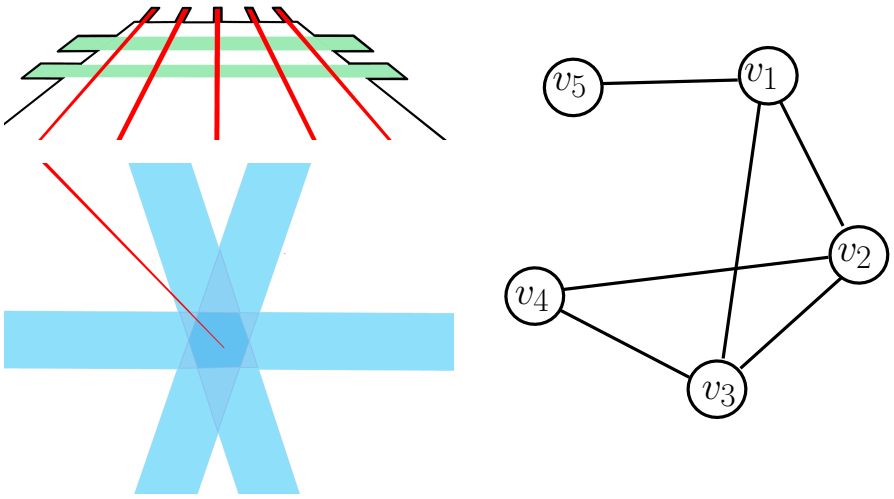


Figure 3.4: On top we have the non cover regions, which are green and pairwise disjoint. We place the point guards that don't correspond to a min vertex cover where the vertex regions and non cover regions intersect. Below is the cover region where the point of the red vertex region ends. Note that the truncated triangles on the boundary look nearly rectangular. To the right we have the graph that we have constructed this polygon from.

For each vertex in G, we modify the top of the boundary of $\mathcal{P}$ to get a truncated triangle which is angled to create a convex region pointing to its corresponding critical point. The truncated triangle's give us the *vertex region*, which is defined as the region of points that can be seen by all points in the truncated triangle. The two angles at the top of the triangle are designated a value that makes the tip of the triangle region "very close" to its corresponding critical point. The critical points are inside the triangle region.
The *cover regions* are defined as the intersection of the vertex region and each edge region which the critical point is contained in.
The *non cover regions* are constructed from rectangles cut into the bottom left and bottom right near the truncated triangles at the top of the polygon. We define these regions as the set of points that can be seen by all points in both the left and right rectangles. For any min vertex cover of size m in G, we will have n-m non-cover regions in order to force a dependency between position of the guards and non-membership of the cover.

Lemma 3.3.1. *Suppose we have a simple graph $G(V, E)$ and $|V| = n \geq 3$. Then G has a min vertex cover $C = \{v_1, \ldots, v_m\}$ where $|C| = m$ if and only if $\mathcal{P}$ has an optimal guard set of size n.*

Proof. Let C be a min vertex cover of size m. As there are n vertex regions that are pairwise disjoint, then we cannot see $\mathcal{P}$ with fewer than n guards. So we need to show that n guards are sufficient to see $\mathcal{P}$. The edge regions can be seen by m guards at the cover regions that correspond to vertices in C. This also covers m many vertex regions, as cover regions are contained in the corresponding vertex regions. The non cover regions can be seen by n-m guards, and we place those guards where they intersect with their corresponding vertex regions. We now see all edge, cover, non cover, and vertex regions. As a single point guard at the critical region covers the main area, we see $\mathcal{P}$.
Conversely, let $\mathcal{P}$ have an optimal guard set of size n. The guard set must see n pairwise disjoint vertex regions, and so a point guard must be located in each vertex region. We must also have n-m guards at the non cover regions to see them. As the non cover regions are

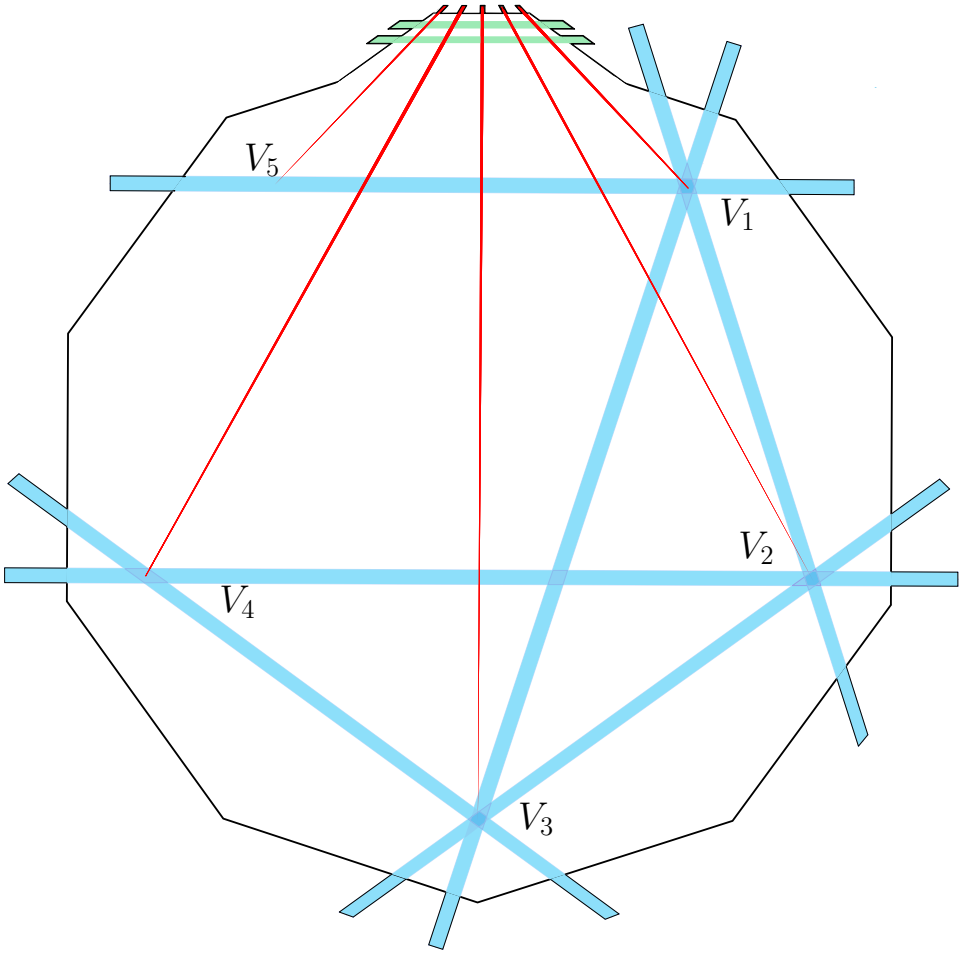


Figure 3.5: A complete construction of our polygon $\mathcal{P}$.

all disjoint from the edge regions, then these guards cannot cover any edge regions. This implies that our m remaining guards must see all edge regions. By the way we constructed the polygon, all edge regions are seen if and only if G has a vertex cover of size greater or equal to m. □

To see that $\mathcal{P}$ can be constructed in polynomial time, consider that we used $O(1)$ corners for each vertex and each edge in G. There are at most $\binom{n}{2}$ edges for any simple graph of n vertices, and so we have $O(n^2)$ corners for $\mathcal{P}$.

3.3.2 Overview of polygon for $\exists\mathbb{R}$ hardness

The proof of $\exists\mathbb{R}$ hardness for the art gallery problem uses a construction of a polygon $\mathcal{P}$ where the following is true:

Let Φ be a formula in ETR-INV defined over n variables $\{x_1, \ldots, x_n\}$ and k equations. Then there exists a polygon $\mathcal{P} := \mathcal{P}(\Phi)$ with corners at rational coordinates which can be computed in polynomial time such that Φ has a solution if and only if $\mathcal{P}$ can be guarded by

$\mathcal{G}(\Phi) := N$ guards. The number N follows from the construction, and will be the size of the optimal guard set.

In the NP-hardness proof we neglected to give a precise description of the coordinates of each corner of $\mathcal{P}$. This was because each guard's position could be perturbed within their given region without changing the size of an optimal guard set. The inherent continuity of ETR does not allow us brush over these details when constructing $\mathcal{P}$. For the proof to work, exact coordinates for all guards and corners must be described. Each guard must be described in reference to both the coordinates of the corners, and the variables for the formula Φ in ETR-INV. Rational coordinates are chosen for each corner so that we can multiply each one by the product of denominators and obtain integer coordinates that are described using a polynomial number of bits. Then, putting $\exists\mathbb{R}$ membership and hardness together, we obtain our desired theorem.

Theorem 3.3.2. *(Abrahamsen et al. [19]) The art gallery problem is $\exists\mathbb{R}$-complete, even the restricted variant where we are given a polygon with corners at integer coordinates.*

The polygon has three areas. One is the main area, where every variable in $X := (x_1, \ldots, x_n)$ is represented by a collection of guards. Then there is a gadget area. The gadgets are closed off from the main area except for a narrow corridor area, which links the two together. A large part of the polygon construction involves coming up with intricate structures that allow for consistency checks between the values of each variable and the set of guards that represent them. This part is technically demanding, and we will skip it. For the most part we will choose to just deal with the question of how each of the equations $x = 1$, $x + y = z$, and $x \cdot y = 1$ are simulated. A picture of what the polygon looks like is presented in Figure 3.6.

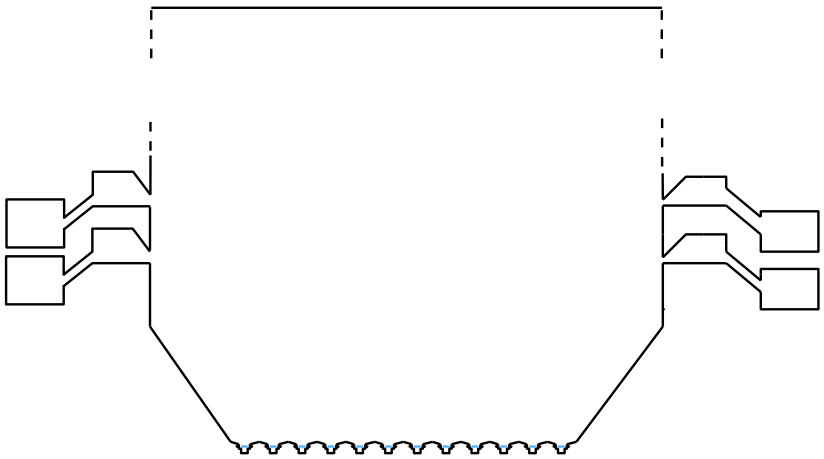


Figure 3.6: A simplified overview of what the polygon looks like. The bottom row is an area for placing guard segments. The corridors and gadgets are attatched on either side. We only describe the gadgets for $x + y = z$, which uses two gadgets combining inequalities on either side, and $xy = 1$, which uses one gadget on the right side.

Each variable $x_i \in X$ is represented by a guard that is placed on what we call a *guard segment*, which are horizontal line segments contained in the interior of $\mathcal{P}$. We define a guard segment $s := ab$, where a is to the left of b. Then s is used to represent a variable x_i by placing some guard g on s. The guard is then specified a position $\frac{1}{2} + \frac{3\|ag\|}{2\|ag\|}$ on s, and g's position

will be a function of the value of x_i.

The idea of the construction is to show that for some solution Φ in ETR-INV, we will get a minimum guard set $\mathcal{G}$ of $\mathcal{P}$ with size $\mathcal{G}(\Phi)$ where the following conditions hold:

- Each variable $x_i \in X$ is specified consistently by $\mathcal{G}$, i.e., there is exactly one guard on each guard segment representing x_i, and all these guards specify the same value of x_i.
- The guard set $\mathcal{G}$ is feasible, i.e., the values of X thus specified is a solution to Φ.

3.3.3 Simulation of $x = 1$

We simulate every value of a single variable in Φ using the structure described in Figure 3.7. Placing these structures into $\mathcal{P}$ enforces that any guard set will require placing a guard on the guard segment $s := ab$. The guards positioned on these segments will be called *segment guards*, and their position on the guard segment will simulate the value of a variable x_i on the interval $[1/2, 2]$. To force a guard onto a precise point we create two wells that can only be seen by a guard at the specified point. The guards positioned on these points will be called *stationary guards*. We can combine these two structures to first force the guard onto the segment, and then use the other one to force the guard onto the point on that segment which corresponds to the guard having value $x = 1$.

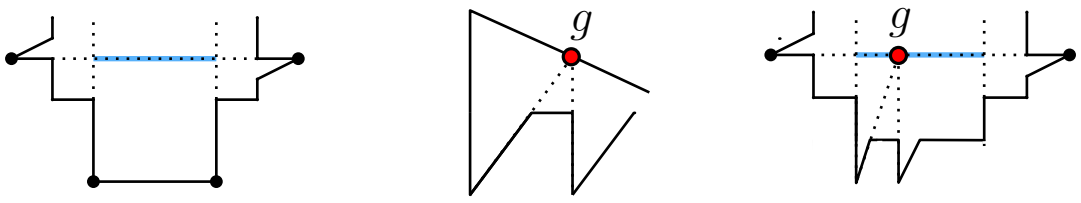


Figure 3.7: On the left, a guard sees the four bolded corners if and only if the guard is on the blue guard segment. This guard is called a segment guard. In the middle, we have an area that is seen by one guard if and only if that guard is located at a precise point. This is called a stationary guard. On the right, we combined the previous two structures to simulate the variable $x = 1$.

3.3.4 Simulation of $x + y = z$

Here we just show a partial description of the gadget for $x + y \geq z$, noting that the other inequality uses a mostly symmetrical gadget on the opposite side of the polygon. Combining both gadgets together gives us $x + y = z$.

Let x_i, x_j, x_l be three variables for a formula $\Phi(X)$ in ETR-INV. We will describe the construction that imposes the inequality $x_i + x_j \geq x_l$. Let $w, v, h > 0$ be rational numbers such that $w > v + 3/2$. Let r_i, r_j, r_l be guard segments of length $3/2$ such that r_j has its left endpoint at $(-w, 0)$, r_i has its right endpoint at $(w, 0)$, and r_l has its left endpoint at $(-2, -h)$. Let $g_i := (w - 2 + x_i, 0)$, $g_j := (-w - 1/2 + x_j, 0)$, and $g_l := (-5/2 + x_l, -h)$ be three guards

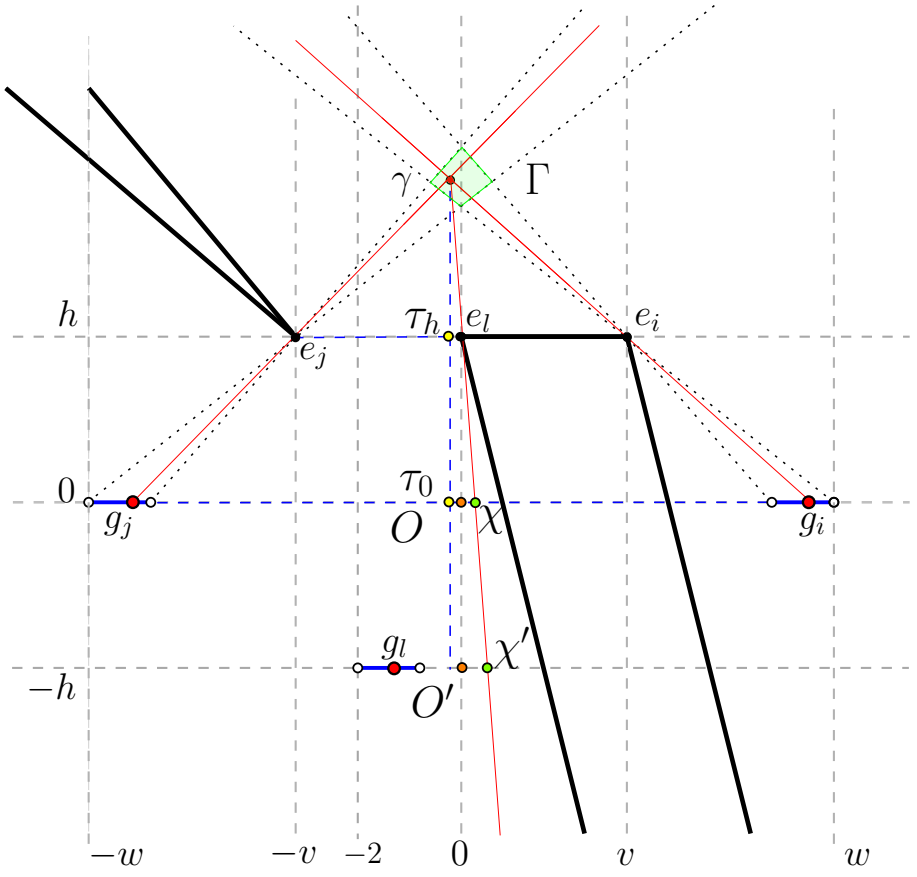


Figure 3.8: Gadget for $x + y \geq z$

on r_i, r_j, r_l, respectively, representing the values $x_i, x_j, x_l \in [1/2, 2]$.

Let $e_i := (v, h), e_j := (-v, h), e_l := (0, h)$. Let Γ be a collection of points γ such that the ray $\overrightarrow{\gamma e_i}$ intersects r_i, and the ray $\overrightarrow{\gamma e_j}$ intersects r_j. Then Γ is a quadrilateral, bounded by the following rays: the rays with origin at the endpoints of r_i and containing e_i, and the rays with origin at the endpoints of r_j and containing e_j. Suppose that

- for every point g_i on r_i and γ in Γ, the points γ and g_i can see each other if and only if γ is on or to the right of the line $\overleftrightarrow{g_i e_i}$.
- for every point g_j on r_j and γ in Γ, the points γ and g_j can see each other if and only if γ is on or to the right of the line $\overleftrightarrow{g_j e_j}$.
- for every point g_l on r_l and γ in Γ, the points γ and g_l can see each other if and only if γ is on or to the right of the line $\overleftrightarrow{g_l e_l}$.

We construct the polygon so that these conditions are true. In the following proposition $y(\gamma)$ is used to denote the y-coordinate of the variable γ. We will use notation to describe x and y coordinates from now on.

Proposition 3.3.1. *(Abrahamsen et al. [19]) g_i, g_j, g_l can together see the whole quadrilateral Γ if and only if $x_i + x_j \geq x_l$.*

Proof. Let $\gamma \in \Gamma$ be the intersection point of the rays $\overrightarrow{g_i e_i}$ and $\overrightarrow{g_j e_j}$. Suppose that the guards g_i, g_j, g_l together see the whole quadrilateral Γ. Since g_i cannot see the area to the left of the line $\overleftrightarrow{\gamma g_i}$, and g_j cannot see the area to the left of the line $\overleftrightarrow{\gamma g_j}$, there are points arbitrarily close to γ which are not seen by any of the guards g_i, g_j. Therefore, g_l has to see γ.
Consider the rays $\overrightarrow{\gamma e_i}, \overrightarrow{\gamma e_j}$, and $\overrightarrow{\gamma e_l}$. Let χ be the intersection point of the ray $\overrightarrow{\gamma e_l}$ with the horizontal line at $y = 0$, and χ' the intersection point of the ray $\overrightarrow{\gamma e_l}$ with the horizontal line at $y = -h$. Note that the guard g_l can see the point γ if and only if g_l is on or to the left of χ'. Recall that two triangles are said to be *similar* if they have corresponding angle pairs and proportional corresponding sides. Consider the line perpendicular to the horizontal axis that passes through the point γ. For this line, we will define τ_h as the point at height h on the line, and τ_0 at the point at height 0 on the line. Then $g_j \tau_0 \gamma$ is similar to $e_j \tau_h \gamma$ and $g_i \tau_0 \gamma$ is similar to $e_i \tau_h \gamma$. Combining these similarities, we see that $\frac{y(\gamma)}{y(\gamma)-h} = \frac{\|g_i - g_j\|}{\|e_i - e_j\|} = \frac{2w + x_i - x_j - 3/2}{2v}$.
From the similarity of triangles $g_j \chi \gamma$ and $e_j e_\ell \gamma$ we get that $\frac{y(\gamma)}{y(\gamma)-h} = \frac{\|\chi - g_j\|}{v}$, and therefore $\|\chi - g_j\| = w + x_i/2 - x_j/2 - 3/4$, and $\chi = (x_i/2 + x_j/2 - 5/4, 0)$. Let $O := (0,0)$ and $O' := (0,-h)$. From the similarity of triangles $O\chi e_l$ and $O'\chi' e_l$ we get that $\chi' = (x_i + x_j - 5/2, 0)$. The condition that the guard g_l is coincident with χ' or to the left of χ' is equivalent to $-5/2 + x_l \leq x_i + x_j - 5/2$, i.e, $x_i + x_j \geq x_l$.

Conversely, if $x_i + x_j \geq x_l$ then the guard g_l is coincident with χ' or to the left of χ', and therefore g_l can see γ. Then the guards g_i, g_j, g_l can together see the whole Γ. □

3.3.5 Simulation of $xy = 1$

The idea of the inversion gadget is to construct a room that requires 3 guards to see it. There are two segment guards. One for a guard representing a variable x, and the other for the guard representing a variable y which is the inverse of x. The other guard is a stationary one, which doesn't represent a variable and is just there to see the rest of the gadget. The segment guards use two regions called a "nook", and an "umbra" that are constructed so that they can only be seen if the two guards are precisely located on the exact point of each segment that represents the variables x and y. They are pictured in figure 3.9 and defined as follows.

Definition 3.3.2. *(nook and umbra)*
Let $\mathcal{P}$ be a polygon with guard segments $r_0 := a_0 b_0$ and $r_1 := a_1 b_1$, where r_0 is to the left of r_1. Let c_0, c_1 be two corners of $\mathcal{P}$, such that c_0 is to the left of c_1. Suppose that the rays $\overrightarrow{b_0 c_0}$ and $\overrightarrow{b_1 c_1}$ intersect at a point f_0, the lines $\overrightarrow{a_0 c_0}$ and $\overrightarrow{a_1 c_1}$ intersect at a point f_1, and that $Q := c_0 c_1 f_1 f_0$ is a convex quadrilateral contained in $\mathcal{P}$.
For each $i \in \{0,1\}$ define the function $\pi_i : r_i \longrightarrow f_0 f_1$ such that $\pi_i(g_i)$ is the intersection of the ray $\overrightarrow{g_i c_i}$ with the line segment $f_0 f_1$, and suppose that π_i is bijective. We say that Q_n is a nook of the pair of guard segments r_0, r_1 if for each $i \in \{0,1\}$ and every $g_i \in r_i$, a guard at g_i can see all of the segment $\pi_i(g_i) f_{1-i}$ but nothing else of $f_0 f_1$. We say that Q_u is an umbra of the segments r_0, r_1 if for each $i \in \{0,1\}$ and every $g_i \in r_i$, a guard at g_i can see all of the segment $\pi_i(g_i) f_i$ but nothing else of $f_0 f_1$. The functions π_0, π_1 are called projections of the nook or the umbra.

Definition 3.3.3. *(critical segment and shadow corners)*
Consider a nook or an umbra $Q := c_0 c_1 f_1 f_0$ of a pair of guard segments r_0, r_1. The line segment $f_0 f_1$ is called the critical segment of Q, and the corners c_0, c_1 are called the shadow corners of Q.

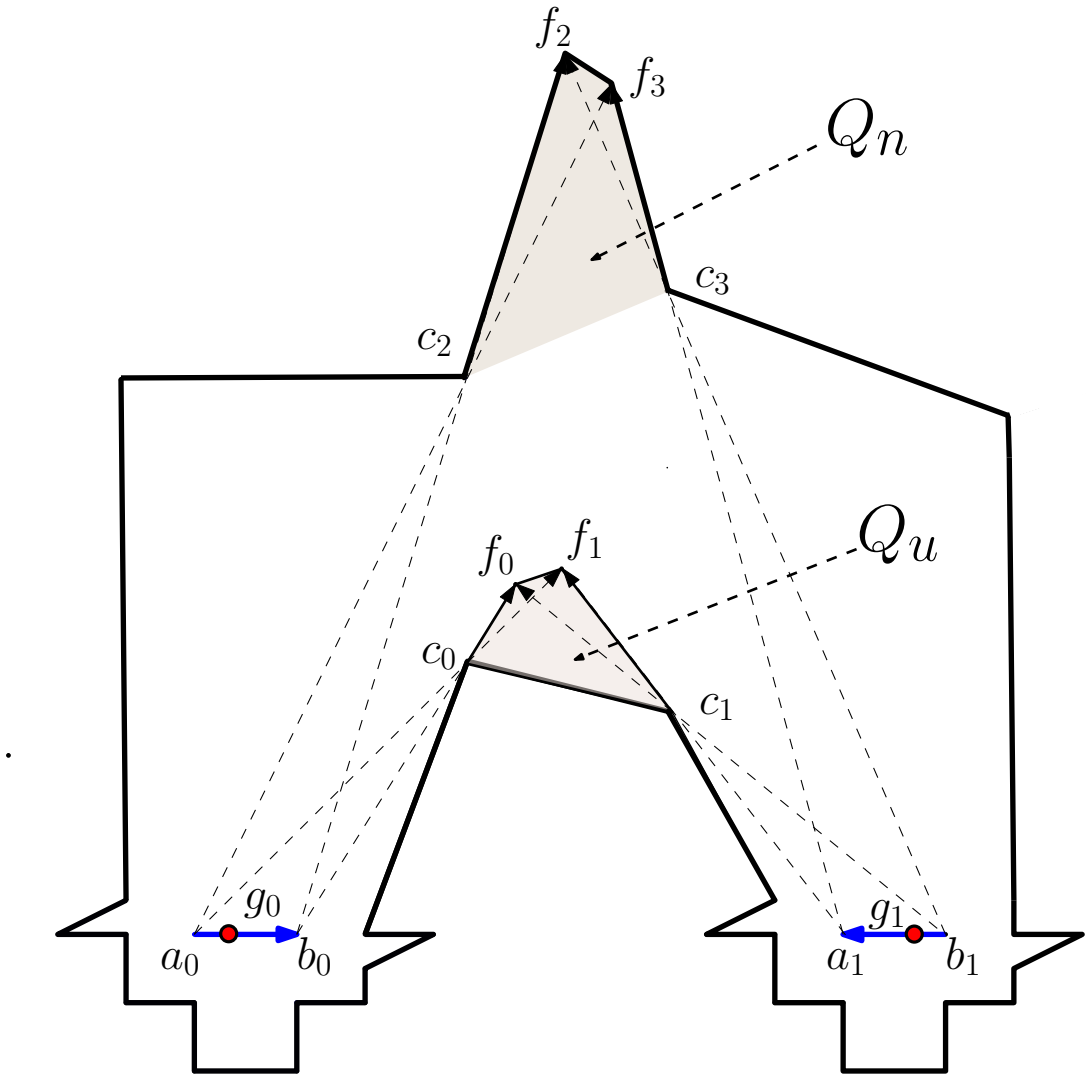


Figure 3.9: overview of the inversion gadget. nooks, umbras, shadow corners and the critical segment are labeled.

Overview of the inversion gadget

Abrahamsen et al. argued that "in order to get rational coordinates of the shadow corners of Q_u and Q_n, it seems necessary to have r_0 and r_1 at different y-coordinates." [19] They also argued that the guard segments must be orientated to have directions. So far we have assigned the guard g representing a variable x as having the value $\frac{1}{2} + \frac{3\|ag\|}{2\|ab\|}$. We will call this a right orientation of the guard segment. A left orientation will assign g to the value $\frac{1}{2} + \frac{3\|bg\|}{2\|ab\|}$.

Our goal here has been, to the best of our ability, to describe their proof without the use of those large fractions. The reader here is given a more expanded description of how the proof works but without saying what the numbers are. To actually check that the proof works requires a program like Maple. We refer to section 4.13.1 of their paper where their proof of the inversion gadget is described in full [19].

Here is how the inversion gadget works. Let r_0 and r_1 be two guard segments representing variables x and y, respectively. We want to construct an umbra Q_u such that if the guards at r_0 and r_1 together see the critical segment f_0f_1 of Q_u, then one of the inequalities $x \cdot y \leq 1$ or

$x \cdot y \geq 1$ follows. The argument here will be to show that wherever the guard g_0 is positioned at on r_0, we can find rational numbers h_l, h_r to define the four corners of the umbra region that allow for a guard g_1 to see the critical segment together with g_0. We construct the nook Q_n in a similar way, choosing rational numbers from the parameters h_l, h_r so that the nook is above the umbra. The guards at r_0 and r_1 will see the critical segment f_2f_3 of Q_n and their visibility regions will also intersect at a single point on f_2f_3. Then the other of the two inequalities follows, so that in effect, $x \cdot y = 1$.
The part of finding rational coordinates for the nook, umbra, and guard segments r_0, r_1 can be thought of as a particular rational valued solution to a parametric equation. Each of the four corners of Q_u and Q_n will be defined using parameters for their y coordinates, which will be h_l, h_r. Abrahamsen et al. were able to do this by using the following values. For guard segments $r_0 := (a_0, b_0)$ and $r_1 := (a_1, b_1)$ they set the values $a_0 = (1/2, 0)$, $b_0 = (2, 0)$, $a_1 = (13.9, 0.1)$, and $b_1 = (15.4, 0.1)$. The shadow corners are defined as $c_0 = c_2 = (7, h_l)$, where the corner is on the left side, and $c_1 = c_3 = (9, h_r)$ when on the right side. Note that $b_1 := 15.4$ corresponds to the value $1/2$ as the guard segment r_1 has a left orientation. Similarly, $a_1 := 13.9$ corresponds to the value 2. So we have the value $g_1 := 15.4 - g_0$. Now we give a step by step process of the umbra construction. Figures 3.10, 3.11, 3.12 loosely correspond to the steps involved.

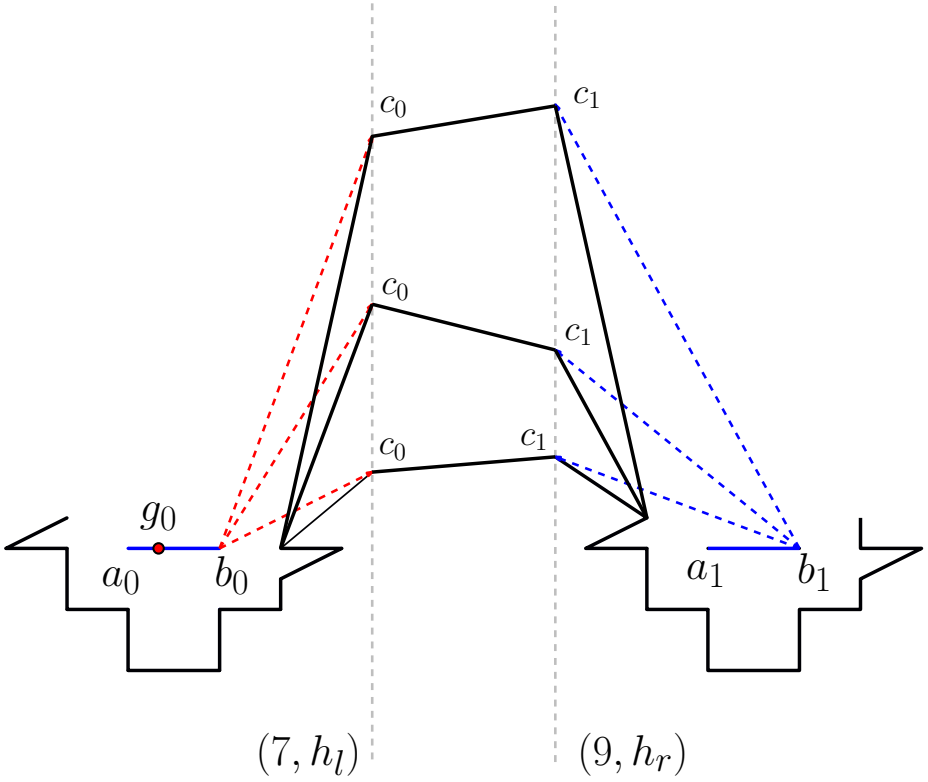


Figure 3.10: The umbra can be placed along the horizontal lines in this picture, at any point where both h_l, h_r are rational and g_0, g_1 will have represent the multiplicate inverses of variables x, y.

Umbra construction

(1) Calculate the parameters for the coordinates of f_0, f_1, which we can do by calculating the intersection point of two lines. A formula for this is described in their paper.

$f_0 = \overrightarrow{b_0c_0} \cap \overrightarrow{b_1c_1}$
$f_1 = \overrightarrow{a_0c_0} \cap \overrightarrow{a_1c_1}$

(2) The parameters for f_0, f_1 now allow us to define the critical segment $\overrightarrow{f_0f_1}$. Given a guard g_0 corresponding to the value of variable x, we calculate e as the intersection of

$e = \overrightarrow{g_0c_0} \cap \overrightarrow{f_0f_1}$.

(3) To find g_1 we take the inverse of its projection, which is calculated as $\pi_1^{-1}(e_0) = \overrightarrow{e_0c_1} \cap \overrightarrow{a_1b_1}$. This evaluates to $\pi_1^{-1}(\pi_0(g_0, 0)) = (\frac{1}{g_0}, 0.1)$. We then have that $x(g_1)$ corresponds to the value $y = 1/x$ placed on the guard segment r_1. Abrahamsen et al. [19]) chose a specific value of $g_0 = 1$ here. Then g_1 must correspond to $1 = \frac{1}{g_0}$. We get

$$g_1 = \frac{\alpha_0 h_l^2 - \alpha_1 h_l h_r + \alpha_2 h_l + \alpha_3 h_r^2 - \alpha_4 h_r}{\alpha_5 h_l - \alpha_6 h_l h_r + \alpha_7 h_l + \alpha_8 h_r^2 - \alpha_9 h_r}$$

where $\alpha_0, \ldots, \alpha_9$ are integer coefficients.

(4) Given our choice of $g_0 = 1$, we can now calculate h_r by solving the equation $15.9 - g_1 = 1$, which is expressed as a quadratic formula:

$$h_r = \frac{\beta_0 h_l + \beta_1 \pm \sqrt{\beta_2 h_l^2 - \beta_3 h_l + \beta_4}}{\beta_0}$$

Where $\beta_0, \ldots, \beta_5$ are specific integer coefficients. Given h_l, h_r must be rational, then this implies the part $\sqrt{\beta_2 h_l^2 - \beta_3 h_l + \beta_4}$ must be the square of a rational number. At this point in the argument, Abrahamsen et al say they used Maple to find such a rational number. So we find some large fractions h_r, h_l that satisify the equation in (5). We are now done constructing the umbra, because we know the coordinates of $Q_u = f_0f_1c_0c_1$.

Lemma 3.3.2. *(Abrahamsen et al. [19]) If guards g_0, g_1 on r_0, r_1 respectively, together see f_0f_1, then $xy \leq 1$.*

Proof. Let $\pi_0 : r_0 \longrightarrow f_0f_1$ and $\pi_1 : r_1 \longrightarrow f_0f_1$ be projections associated with Q_u. Since g_0 represents the variable x, then we have $g_0 := (x, 0)$. Let

$$e := \pi_0(g_0) = \overrightarrow{g_0c_0} \cap \overrightarrow{f_0f_1}$$

Now, $\pi_1^{-1}(e) = (15.9 - 1/x, 1/10)$, which represents the value $15.9 - (15.9 - 1/x) = 1/x$ on r_1. In order to see f_0f_1 together with g_0, the guard g_1 has to stand on or to the right of $\pi_1^{-1}(e)$. This corresponds to y being at most $1/x$. In other words, if both guard g_1 on r_1 sees f_0f_1 together with g, then $xy \leq 1$. □

Nook construction

For the nook construction, we follow a similar procedure to how we constructed the umbra. Just as the corners f_0, f_1 were defined using the parameters h_l, h_r, we do the same for f_2, f_3. Similarly the shadow corners are defined as $c_2 = (7, h_l)$, and $c_3 = (9, h_r)$ where we keep the x-coordinates the same as in c_0, c_1. From here, we repeat steps (2)-(5), except we pick different values for the parameters h_l, h_r. In particular, we assign values for h_l, h_r such that

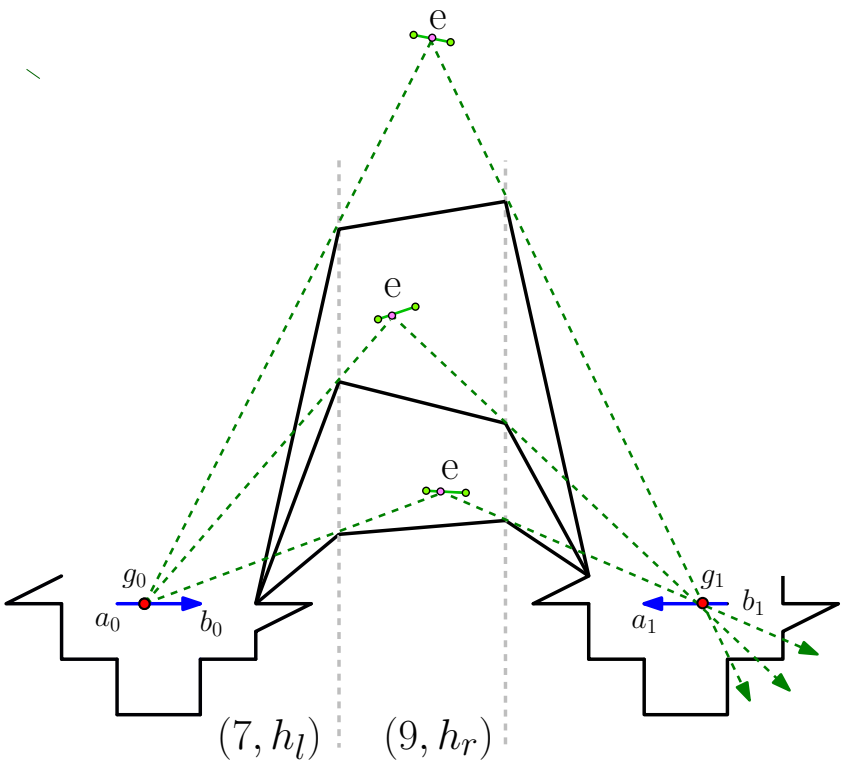


Figure 3.11: The green lines show the projection of the guard onto the critical segment, and the inverse projection onto the position of g_1.

all corners of the nook must be located above all corners of the umbra so as to avoid the regions from intersecting each other. Then we can prove the following lemma.

Lemma 3.3.3. *(Abrahamsen et al. [19]) If guards g_0, g_1 on r_0, r_1 respectively, together see f_2f_3, then $xy \geq 1$.*

Proof. Let $\hat{\pi_0} : r_0 \longrightarrow f_2f_3$ and $\hat{\pi_1} : r_1 \longrightarrow f_2f_3$ be projections associated with Q_n. Since g_0 represents the variable x, then we have $g_0 := (x, 0)$.

$$\hat{e} := \hat{\pi_0}(g_0) = \overrightarrow{g_0c_2} \cap \overrightarrow{f_2f_3}$$

Now, $\hat{\pi_1}^{-1}(\hat{e}) = (15.9 - 1/x, 1/10)$, which represents the value $15.9 - (15.9 - 1/x) = 1/x$ on r_1. In order to see f_2f_3 together with g_0, the guard g_1 has to stand on or to the left of $\hat{\pi_1}^{-1}(e)$. This corresponds to y being at most $1/x$. In other words, if both the guard g_1 on r_1 sees f_2f_3 together with g_0, then $xy \geq 1$. □

Putting the previous two lemmas together gives us the equality.

Lemma 3.3.4. *(Abrahamsen et al. [19]) If guards g_0 and g_1 placed on guard segments r_0 and r_1, respectively, see both critical segments f_0f_1 and f_2f_3, then the corresponding values specified by g_0 and g_1 satisfy $xy = 1$.*

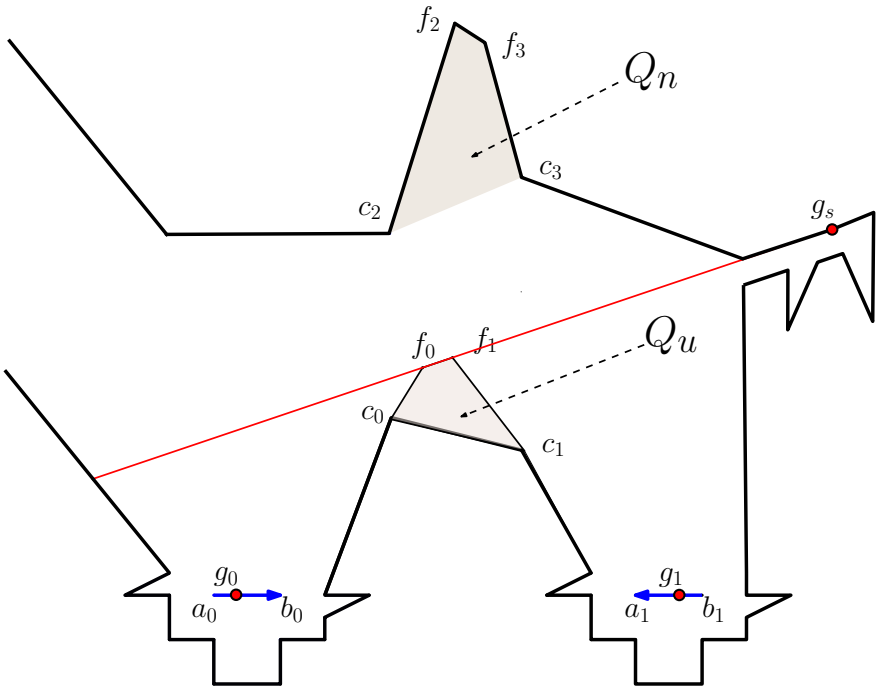


Figure 3.12: Our final construction of the inversion gadget

3.4 All algebraic numbers as guards

Recall from the proof of Theorem 3.2.1 that we scaled each variable x_i down by a constant $\varepsilon = 1/8^m$. After shrinking each variable we then shifted them forward on the real line by adding $7/8$ to each variable giving us an instance $\Psi(Y)$ where each variable $y_i = \frac{x_i}{8^m} + 7/8$. Though we did not give a complete proof of the last few reductions, the steps involved in those reductions do not prevent us from generalizing the values of $1/8^m$ and $7/8$ to some arbitrary rational numbers, which is stated in the following lemma.

Lemma 3.4.1. *(Abrahamsen et al. [19]) Let Φ be an instance of ETR with variables $x_1, \ldots, x_n$. Then there exists an instance Ψ of ETR-INV with variables $y_1, \ldots, y_m$, where $m \geq n$ and constants $c_1, \ldots, c_n, d_1, \ldots, d_n \in \mathbb{Q}$, such that*

- there is a solution to Φ if and only if there is a solution to Ψ, and
- for any solution $(y_1, \ldots, y_m)$ to Ψ, there exists a solution $(x_1, \ldots, x_n)$ to Φ where $y_1 = c_1x_1 + d_1, \ldots, y_n = c_nx_n + d_n$.

This can be combined with Theorem 3.3.2 to get a corollary.

Corollary 3.4.1. *(Abrahamsen et al. [19]) Let Φ be an instance of ETR with n variables. Then there is an instance $(\mathcal{P}, g)$ of the art gallery problem, and constants $c_1, d_1, \ldots, c_n, d_n \in \mathbb{Q}$, such that*

- if Φ has a solution, then $\mathcal{P}$ has a guard set of size g, and
- for any guard set $\mathcal{G}$ of $\mathcal{P}$ of size N, there exists a solution $(x_1, \ldots, x_n) \in \mathbb{R}^n$ to Φ such that $\mathcal{G}$ contains guards at positions $(c_1x_1 + d_1, 0), \ldots, (c_nx_n + d_n, 0)$.

We are now ready to show that there are infinitely many rational polygons that require irrational guards for an optimal guard set.

Definition 3.4.1. *(real algebraic numbers). A real algebraic number α is defined as a real number that is a root of some non-zero univariate polynomial with rational coefficients.*

There are infinitely many irrational algebraic numbers.

Theorem 3.4.1. *(Abrahamsen et al. [19]) Given any real algebraic number α, there exists a polygon $\mathcal{P}$ with corners at rational coordinates such that in any optimal guard set of $\mathcal{P}$ there is a guard with an x-coordinate equal α.*

Proof. Let $P(x)$ be a polynomial of degree more than 0 in one variable x such that the equation $P(x) = 0$ has α as a solution. Remember that there are finitely many real roots of a finite degree polynomial. This means we can choose integers p_1, p_2, q_1, q_2 such that α is the only solution in the interval $[p_1/q_1, p_2/q_2]$. Then the formula $P(x) = 0 \wedge p_1 \leq q_1 x \wedge q_2 x \leq p_2$ is a formula in ETR with a unique solution $x = \alpha$. Now, by Corollary 3.4.1, there exists a polygon $\mathcal{P}$ and rational constants c, d such that in any optimal guard set of $\mathcal{P}$, one guard has coordinates $(c\alpha + d, 0)$. By subtracting d from the x-coordinate of all corners of $\mathcal{P}$ and then dividing all coordinates by c, we get a polygon $\mathcal{P}'$ such that any optimal guard set of $\mathcal{P}'$ has a guard at the point $(\alpha, 0)$. □

This may seem like bad news for finding an efficient algorithm that decides the size of an optimal guard set. But in the next chapter we will explore the possibility that requiring irrational guards for optimal guard sets is a rare occurance in the AGP.

Chapter 4

A discussion on smoothed analysis, a practical algorithm for the AGP and the chromatic art gallery problem

4.1 A discussion on smoothed analysis

In complexity theory we traditionally define the running time of algorithms by their worst case scenario of any input. This is not the only approach that can be taken. Another approach is to consider taking the average running time of an instance that is selected with uniform randomness from all possible instances. However, there are problems with this approach.

- The first is we need to define a probability space. Suppose we take a large selection of vertices for our polygon and randomly choose coordinates in a closed space in $\mathbb{R}^2$ for our vertices, then draw edges between them. Then we will likely get two edges intersecting, and our polygon will not be simple. So to begin with, selecting well-defined polygons at random requires a lot of extra assumptions in order to even know how to select such instances with uniform randomness.
- In real world cases we may want to look at polygons that have specific properties. For example, the probability of encountering three collinear points on the plane when their coordinates are randomly generated in some bounded space is zero. But in the real world, we encounter collinearities all the time. So we might only be interested in small areas of a given probability space.
- Polygon inputs that cause high running times for a given algorithm could all share a similar structure. What this means is that a specific area of a given probability space could contain a much higher density of polygons with optimal guard sets that require irrational guards, for example. An average case analysis might give us a low expected running time, but we want to make sure that our average running time is low for any arbitrarily chosen area of the probability space. In other words we may want our worst case scenario polygons to not all look similar to each other.

The input model in a smoothed analysis consists of two steps. In the first step, an adversary specifies an arbitrary input. After that, in the second step, this input is slightly perturbed at random. We now define smoothed running time and explain how it deals with these issues. First we let $\delta > 0$ be a bound on the amount that an instance can be perturbed.

For a given instance I, we will define the probability space $(\Omega_\delta, \mu_\delta)$ where each $x \in \Omega_\delta$ defines some perturbed instance of I, which we will denote as I_x. The symbol μ_δ denotes the expected run time in our specified probability space. We denote $T(I_x)$ as the time to solve the instance I_x. We now define the smoothed expected running time of an *instance* I as

$$T_\delta(I) = \mathop{\mathbb{E}}_{x \in \Omega_\delta} T(I_x) = \int_{x \in \Omega_\delta} T(x)\mu_\delta(x)$$

The set of instances of size n is defined as Γ_n. We define the smoothed running time as

$$T_{\text{smooth}}(n) = \max_{I \in \Gamma_n} \mathop{\mathbb{E}}_{x \in \Omega_\delta} T(I_x)$$

Let us now see how this deals with the issues of average case analysis. If there are types of polygons where itself, and 'similar' polygons, (i.e slightly perturbed ones) have high running times, then this will show up in the smoothed running time, which will be high, because polygons similar to our input make up the area of our probability space. This would not be detected in our average case analysis, because the more δ increases, the less similar our perturbed polygon will be to I.
If however we get a low smoothed running time, then it implies that not only is every single polygon with a high running time a fairly isolated incident, i.e, it is *rare* to encounter an instance with a high running time, but also that the set of polygons with high running times do not look similar to each other.
If the area of the probability space is one that we are interested in for real world applications, then a low smoothed running time will tell us that an algorithm has practical use in the real world as well. If we are interested in a particular real world polygon, then we can be assured that any similar polygon will often have a low running time for the algorithm.
Dobbins et al. [10] recently examined the AGP from a smoothed analysis perspective. They compared different types of peturbations, and the running times that each one gave. We describe the vertex perturbation example. In this scenario we have an input $\mathcal{P}_n$, which is a polygon with vertices $\{v_1, \ldots, v_n\}$. For $\delta > 0$ we place a ball $B(v_i, \delta)$ around each vertex. Then our peturbed instance of P_n is the polygon defined by n random variables $\{v'_1, \ldots, v'_n\}$ where $\|v'_i - v_i\| < \delta$. So the probability space is defined as $\Omega_\delta = \text{disk}(\delta)^n$. Here δ must be chosen small enough so that the polygon is still simple, i.e, its boundary doesn't cross for some perturbed instance. This is an example of where we need to figure out how to define a probability space carefully in order to do smoothed analysis. We skip the details of this here, but we note that from their paper that a smoothed running time of

$$T_{\text{smooth}}(n) = O\left(\log\left(\frac{nL}{\delta}\right)\right)$$

where L is an integer was acheived for some of the cases they examined.
Choosing how to perturb the polygon can depend on the interests of the researcher. For example, we can perturb the vertices as is shown in Figure 4.1. Or we can slightly stretch the edges of $\mathcal{P}$. A difference between these two choices is that the former removes collinearities, while the second does not. If for example, our algorithm has a high running time for certain polygons with collinearities, then our smoothed running could be higher for the edge stretching model.

4.2 A practical algorithm for the art gallery problem

In this section we look at a practical algorithm that perturbs the visibility regions of each guard. Suppose we let $\delta > 0$, and g be a guard in some guard set $\mathcal{G}$. Now let $\overline{gp}$ be a

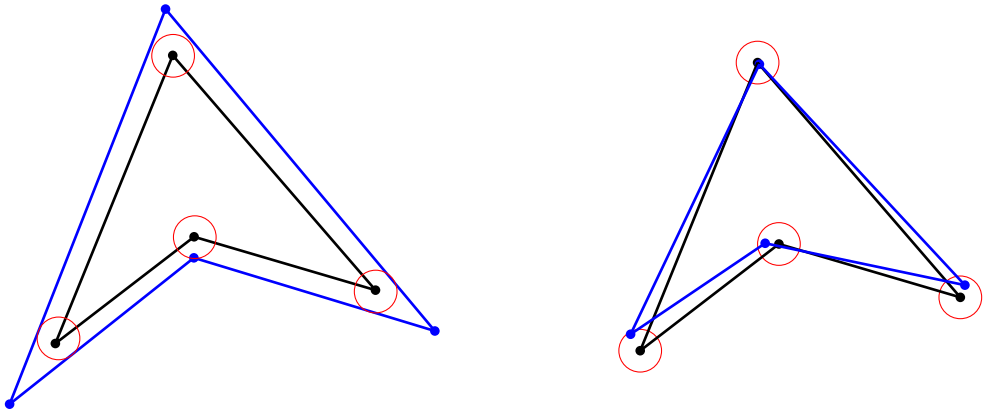
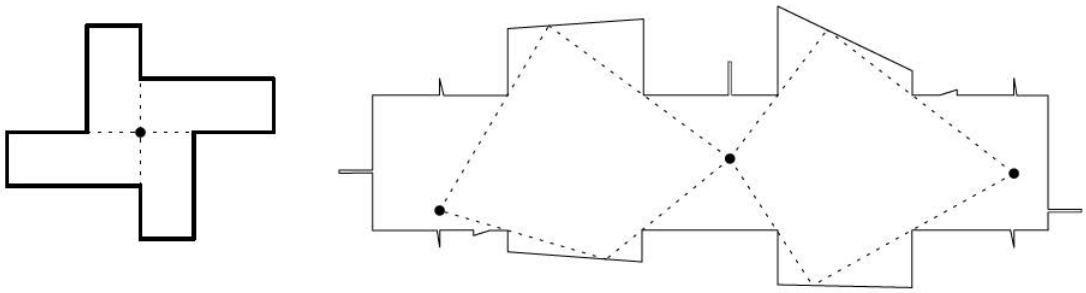


Figure 4.1: On the left is a polygon input which we have perturbed by stretching the edges. This would not break any prexisting collinearities. However for the polygon on the right we have randomly perturbed the vertices to a close by position. Under this model of smoothed analysis, the probability of obtaining collinear edges is zero. If our algorithm were to have a high running time because of collinearities, as is often the case in computational geometry, then the model on the right has worst case running times that are fragile to the perturbation, unlike edge stretching.

maximal line segment that intersects (i.e, is collinear with) some reflex vertex r. We define $A(g, r, \delta)$ as the region obtained by rotating $\overline{rp}$ clockwise and counterclockwise by an angle δ, then taking union of the maximal segments contained in that angle of rotation. Now, let $A(g, R, \delta)$ be all such regions obtained by taking the union of $A(g, r, \delta)$ for all reflex vertices in $V(g)$. Then we define the *δ-visibility enhancing region* $V_\delta(g)$ as $V(g) \cup A(g, R, \delta)$. We define the *$\delta$-diminished visibility region* as $V_{-\delta}(g) = V(g) \backslash A(g, R, \delta)$.

Given a polygon $\mathcal{P}$, we say that a set $\mathcal{G}_\delta$ is a *δ-guard set* of $\mathcal{P}$ if $\bigcup_{g \in \mathcal{G}} V_\delta(g) = \mathcal{P}$. We denote $\text{opt}(\mathcal{P}, \delta)$ as the size of the minimum δ-guarding set. We say that a polygon $\mathcal{P}$ is *vision-stable* or equivalently has *vision-stability* $\delta > 0$, if $\text{opt}(\mathcal{P}, -\delta) = \text{opt}(\mathcal{P}, \delta)$. The idea then, is to try and approach the optimal guard set number from both the perspective of enhanced vision and diminished vision by iteratively decreasing the value of δ. Hopefully, we will in a finite number of steps find a δ vision stable polygon. If our minimal guard set is size k for diminished visibility, then we can certainly guard the polygon using the same number of guards with normal, or enhanced vision. If our guard set with enhanced vision is the same size as the guard set for diminished visibility guard sets, then we know that we arn't going to find a smaller guard set no matter how many times we divide δ in two.

Hengeveld and Miltzow [24] developed an efficient algorithm that decides the optimal guard set for polygons that have vision stability. The justification is that they think vision stable polygons are *typical* instances of polygons. Consider the two polygons in Figure 4.2. The first one is not vision stable because of some collinearities. It is clear that two guards are required for $V_{-\delta}(g)$ given any value of $\delta > 0$. But in computational geometry there are methods for dealing with high running times that result from collinearities anyway. The second polygon is one that was constructed in order to prove that irrational guards are sometimes required for an optimal guard set (Abrahamsen et al. [9]). This polygon, according to Hengeveld and Miltzow, "took decades of research...as of this writing no second similar polygon is known." So they think that such polygons are rare in practice.



(Hengeveld and Miltzow. [24])

Figure 4.2: On the left is a polygon that has an optimal guard set of size one. For any $\delta > 0$, we will need more than one guard for δ-diminished visibility. Hence the polygon is not vision stable. The polygon on the right requires a guard to be located on an irrational coordinate, so is not vision stable either.

The vision-stable algorithm

Their algorithm uses the following observations. The closer a guard is to a reflex vertex in $\mathcal{P}$, the more its corresponding visibility region changes. This observation motivates choosing an arrangement $\mathcal{A}$ of regions that generate a candidate set and witness set. Each set becomes more sparse the further away we move from reflex vertices (see Figure 4.4). It is not our wish to give a formal proof and description of how the algorithm works, as this requires introducing several new concepts. So we will give a superficial, and informal description of how this works.
The algorithm starts by shooting rays out from each reflex vertex where the interior angle between each ray is some chosen value of δ. We consider the segments of the ray that are inside $\mathcal{P}$. Each intersection point of the rays, and each region bounded by these rays is used to define the candidate set and witness set. Both sets are found using methods from linear programming. If the polygon has vision stability for $\delta > 0$, then we know the size of the optimal guard set. Otherwise, the algorithm divides δ in two and preprocesses a finer arrangement. There are $O(\frac{1}{\delta})O(r) = O(\frac{r}{\delta})$ many rays shot out for each preprocessing of the arrangement.

This algorithm then terminates once it has found an optimal guard set, i.e, when $\text{opt}(\mathcal{P}, -\delta) = \text{opt}(\mathcal{P}, \delta)$. Importantly, this implies that the algorithm only ever terminates for polygons that

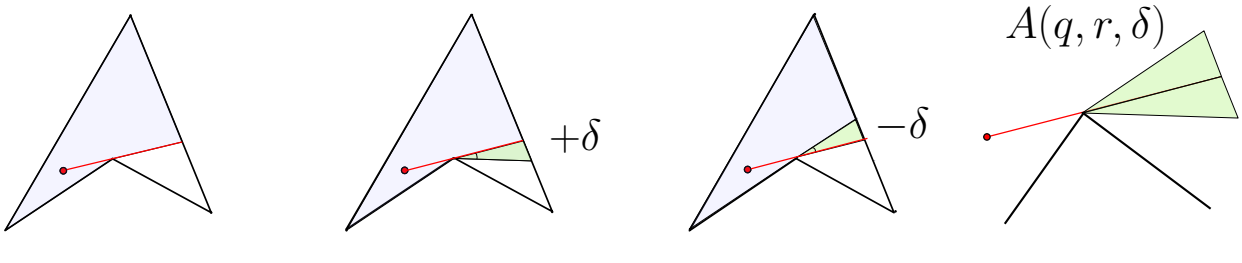


Figure 4.3: On the left, we have our standard visibility. In the middle two pictures we have δ-enhanced and δ-diminished visibility regions. The picture on the right shows the region that is obtained from taking maximal line segments rotated around r at angles less than or equal to δ.

are vision stable. But it can never return an incorrect answer, and if vision stable polygons are typical on all inputs of size n, which seems likely, then we have an algorithm that will terminate with high probability and has a much lower running time. Hengeveld and Miltzow also found that this algorithm approximated the coordinates of the guards used for the guard set in the polygon requiring an irrational guard, in Figure 4.2.

4.3 The chromatic art gallery problem

Suppose $\mathcal{G}$ is a guard set of a polygon $\mathcal{P}$. We say that we have a *guard colouring* of $\mathcal{P}$ if for any two guards in $\mathcal{G}$ with intersecting visibility regions, the guards are assigned different colours. We denote $C(\mathcal{G})$ as the minimum number of colours needed for a guard colouring of $\mathcal{G}$. Let $T(\mathcal{P})$ be the set of all guard sets of $\mathcal{P}$. Then we can define the chromatic guard number of $\mathcal{P}$, which we define as $\chi_{\mathcal{G}}(\mathcal{P})$ where

$$\chi_{\mathcal{G}}(\mathcal{P}) := \min_{\mathcal{G} \in T(\mathcal{P})} C(\mathcal{G}).$$

The chromatic art gallery problem (CAGP) is the problem of finding the chromatic guard number of a polygon $\mathcal{P}$. This problem was introduced in 2010 by Erickson and LaValle [25], and has some interesting real world applications for wireless communications. Wireless coverage can be represented as visibility regions. Two signals must operate on different frequencies, and the goal is to minimize the number of frequencies that are used. This corresponds to finding the chromatic guard number of a polygon. More about this application is discussed here [26].
Though several variants of the chromatic art gallery problem have been examined in terms of complexity theory, we have not found any papers that have examined this problem from the context of the existential theory of the reals.

4.3.1 NP-hardness of CAGP

The chromatic art gallery problem was proven to be NP-hard in 2014 (Fekete et al. [27]), using an elegant and very short proof. Their proof uses a reduction from the problem of finding a minimum point cover for a set of lines on the plane, which is defined as follows.

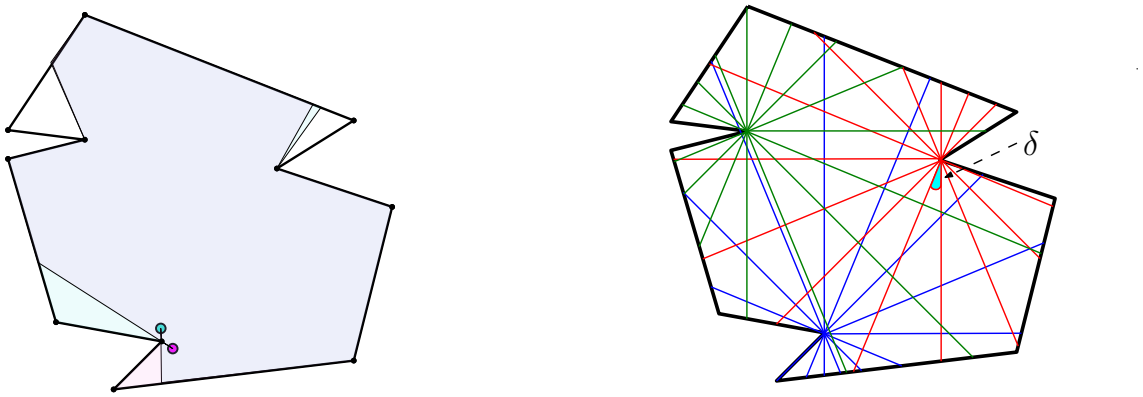


Figure 4.4: On the left we have a picture displaying two guards close to a reflex vertex. Small changes in the position of a guard affects their visibility regions more when they are closer to reflex vertices. On the right we have our preproccessed witness-guard set.

Definition 4.3.1. *(Minimum point cover) Consider a set $P = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ on the plane. A minimum point cover is the smallest set of lines $\mathcal{L} = \{\ell_1 \ldots, \ell_k\}$ such that every point in P is on some line $\ell_j \in \mathcal{L}$*

Finding the minimum point cover is NP-hard (Megiddo and Tamir [28]). We think their proof works for the general art gallery problem as well, as their polygon's optimal guard set is the same size as the chromatic guard number. We found the same to be true for our polygon and give a proof of this fact. The following proof adjusts the reduction made in Lemma 3.3.1.

Proposition 4.3.1. *The chromatic art gallery problem is NP-hard*

Proof. Consider the polygon $\mathcal{P}$ from Figure 3.5 in the NP-hard proof of the general art gallery problem. For a graph G with n vertices, the polygon requires n guards. It is easy to see that the chromatic guard number of $\mathcal{P}$ is at most the size of an optimal guard set of $\mathcal{P}$. As the guards in the optimal guard set are all in the vertex regions, then all of their visibility regions intersect. So each guard must be coloured differently. Then the chromatic guard number is at least the size of the optimal guard set, so $\chi_G(\mathcal{P}) = n$. So for the input $G(V, E)$ where $|V| = n \geq 3$, then the graph G has a min vertex cover of size m if and only if $\mathcal{P}$ has chromatic guard number of size n. □

Perhaps this can be strengthed into a $\exists\mathbb{R}$-hard proof using the polygon that Abrahamsen et al. [19] constructed. We will not attempt that here though, as we only described a very small part of their polygon and so a proof would not convince the reader. Another obvious but important fact that can be oserved from Figure 4.2 is that a minimal colouring of a guard set sometimes requires irrational guards in the guard set.

4.3.2 The chromatic art gallery problem is decidable

Proposition 4.3.2. *The chromatic art gallery problem is decidable by encoding the problem as a formula in ETR (Renwick).*

Using Φ to encode the problem

Recall the line arrangement $\mathcal{L}$ from chapter 3 that was discussed when constructing the formula Φ which encoded the AGP as a formula in ETR. Every edge in $\mathcal{P}$ had a line passing through it, and every guard had a line passing through it and every vertex in $\mathcal{P}$. Luckily, this arrangement that has already been used for Φ gives us all of the necessary lines needed to encode the chromatic art gallery problem as a formula in ETR. We will denote the formula for the CAGP as Ω. As every formula of ETR is decidable, then simply encoding Ω as a formula in ETR will be enough to prove Proposition 4.3.2. The formula we construct does not have a polynomial bound on its length, and so we were unable to show that computing the chromatic guard number of a polygon is in $\exists\mathbb{R}$.
In the definition of the chromatic guard number, we have to find the minimum colouring for all guard sets. However this does not mean we must construct the formula Ω for arbitrarily large $k \in \mathbb{N}$, or m. We make the following observation.

Observation 4.3.1. *The chromatic guard number of any given polygon is less than or equal to $\lfloor \frac{n}{3} \rfloor$*

Proof. From Fisk's proof of the art gallery theorem, we can observe that placing $k = \lfloor \frac{n}{3} \rfloor$ guards, one per triangle, will give us a guard set of $\mathcal{P}$. We can assign a unique colour for each guard. Therefore we can find a colouring $C(\mathcal{G}) = k$ for $\mathcal{P}$, so the chromatic guard number will be less than or equal to k. □

This gives us a fixed bound on the number of ways of partitioning k visibility regions into m cells. The idea behind the construction of the sentence Ω will be to check each partition if any of the m cells contain intersecting visibility regions. If there are none, then we will be able to achieve an m colouring The following lemma gives us an immediate corollary that any edge of a visibility region is contained in a line from $\mathcal{L}$.

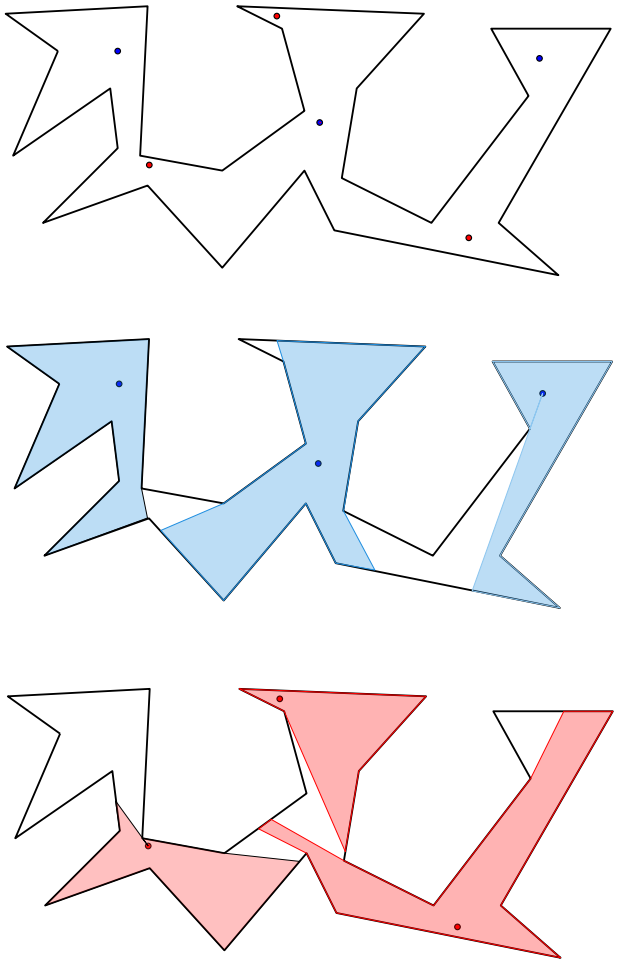


Figure 4.5: At the top, a guard set of size 6. The next two pictures show their 6 visibility regions partitioned into two sets of pairwise disjoint visibility regions. This gives a 2 colouring for the guard set.

Lemma 4.3.1. *Let $V(g)$ be a visibility region of some polygon $\mathcal{P}$. Then the edges of $V(g)$ are either collinear with a reflex vertex of $\mathcal{P}$, or are collinear with an edge of $\mathcal{P}$.*

Proof. Let $e = vu$ be an edge of $V(g)$. Consider the tautology that either e is collinear with g or it isn't. We will show that if e is collinear with g, then e is collinear with a reflex vertex of $\mathcal{P}$, and if e is not collinear with g, then e is collinear with an edge of $\mathcal{P}$.
Case 1: Suppose for contradiction that e is collinear with g and and e is not collinear with a reflex vertex of $\mathcal{P}$. Then the maximal line segment $\overline{gp}$ can be rotated about the apex g by some abitrarily small angle without intersecting $\partial\mathcal{P}$. This implies g can see points on either side of e, contradicting that e is an edge of $V(g)$. So e is collinear with a reflex vertex of $\mathcal{P}$.
Case 2: Suppose e is not collinear with g. As every point $p \in e$ is the boundary point of a maximal line segment $\overline{gp}$, then all p are in $\partial\mathcal{P}$. As e contains two or more points in an edge of $\partial\mathcal{P}$, this implies e is collinear with an edge of $\mathcal{P}$. □

We obtain the following useful corollaries.

Corollary 4.3.1. *Let e be an edge in $V(g)$, the visibility region of some guard g. Then e is contained in a line from $\mathcal{L}$.*

Corollary 4.3.2. *Let g_r, g_s be two distinct guards in a polygon $\mathcal{P}$. Then the visibility regions $V(g_r), V(g_s)$ intersect if and only if they have a pair of edges that intersect, where each edge is contained in a line in $\mathcal{L}$*

We are almost ready to encode Ω. We need to be careful in regards to particular cases here. It may be that the intersection of two visibility regions is contained in some line from $\mathcal{L}$, i.e, for two guards g_r, g_s we have $V(g_r) \cap V(g_s) \subseteq \ell_i$ for some $\ell_i \in \mathcal{L}$. See Figure 4.6 for an example. We want there to exist at least two well defined, non parallel lines, with an intersection point $X^{ij} \in V(g_r) \cap V(g_s)$ whenever g_r, g_s have intersecting regions.

Lemma 4.3.2. *Let g_r, g_s be two distinct guards in a polygon $\mathcal{P}$. Then the visibility regions $V(g_r), V(g_s)$ intersect if and only if there exists a pair of edges that intersect, and those edges are contained in two lines $\ell_i, \ell_j \in \mathcal{L}$ that have an intersection point X^{ij} which sees both guards g_r, g_s*

Proof. Let g_r, g_s be two distinct guards in a polygon $\mathcal{P}$. If $V(g_r) \cap V(g_s) \nsubseteq \ell_i$, for any line $\ell_i \in \mathcal{L}$, then by Corollary 4.3.2 there are two edges that intersect and the lines passing through them are well defined, non parallel and so have an intersection point which sees g_r, g_s. Now let g_r, g_s be two guards where $V(g_r) \cap V(g_s) \subseteq \ell_i$. Firstly, note that $V(g_r) \cap V(g_s)$ cannot be contained in some edge $e \in \partial\mathcal{P}$. If this were the case, then to either side of e, there is no point that both g_r and g_s can see. But the boundary of $\mathcal{P}$ needs to be self intersecting for this to be true. If it isn't, then we can rotate the line segments $\overline{g_r g_s}$ and $\overline{g_s g_r}$ by a small angle around apexs g_r and g_s respectively, without intersecting the boundary. This would imply they both see a point $p \in \mathcal{P}$ that is not contained in ℓ_i. So we contradict the assumption that $V(g_r) \cap V(g_s) \subseteq \ell_i$.
Now that we know that $V(g_r) \cap V(g_s)$ is not contained in an edge of $\mathcal{P}$, we note that $\overline{g_r g_s}$ must intersect some reflex vertex of $\mathcal{P}$. Otherwise, we can rotate the line segments to see a common point p not in ℓ_i, as was just argued. We obtain the same contradiction. So a reflex vertex is contained in $V(g_r) \cap V(g_s)$, and the two edges in $\mathcal{P}$ that the vertex is incident to have lines going through them. The intersection piont of these lines is the reflex vertex.

Conversely, suppose there is a pair of edges from $V(g_r), V(g_s)$ that are contained in two distinct lines $\ell_i, \ell_j \in \mathcal{L}$ respectively, and have an intersection point X^{ij} which sees g_r, g_s. Then $X^{ij} \in V(g_r) \cap V(g_s)$ and so these visibility regions intersect. □

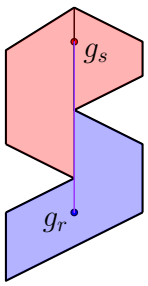


Figure 4.6: Two visibility regions $V(g_r)$, $V(g_s)$ where their intersection is contained in a single line.

Constructing Ω

In the construction of Φ that tested for existence of a guard set size k, the variables x_{ij}, y_{ij} were encoded as the coordinates for each intersecting pair of lines $\ell_i, \ell_j \in \mathcal{L}$. Now all we need to do is encode, for each pair of guards, if an intersection point of a pair of well defined, non parallel lines is seen by both of them. We use a familiar looking predicate to do this and define each guard $g_t := (x_t, y_t)$.

$$\varphi := \text{INSIDE-POLYGON}\left(x_{ij}, y_{ij}\right) \implies \Big[\left[\text{SEES}\left(x_r, y_r, x_{ij}, y_{ij}\right)\right] \wedge \left[\text{SEES}\left(x_s, y_s, x_{ij}, y_{ij}\right)\right] \Big]$$

For each intersection point, we can encode this formula for a pair of guards. Now we know how to test if two visibility regions intersect. From now on, let us abbreviate the above formula as $\varphi_{(s,r)}$ for each pair of guards g_s, g_r.
Fix some k and let $\Gamma := \{C_1, \ldots, C_m\}$ be a partition of k many visibility regions into m nonempty cells. There are $S(k,m)$ ways to do this where $S(k,m)$ denotes stirling numbers of the second kind. Now, for any $C_p \in \Gamma$ we have $|C_p|$ many guards. We want to be able to say that none of the guards in each C_p have intersecting visibility regions. This corresponds to being able to paint the visibility regions in C_p with a single colour. If all of them can be, then we have found a guard set that is m colourable. So, for each pair of visibility regions $V(g_r), V(g_s)$, the following formula is true if and only if $V(g_r) \cap V(g_s) = \emptyset$.

$$\neg \left[\bigvee_{(i,j)\in\{1,\ldots,q\}^2} \varphi_{(s,r)} \right]$$

There are $\binom{|C_1|}{2} + \cdots + \binom{|C_m|}{2}$ pairs of guards that must be checked to decide that our partition Γ is m colourable. So we will end up having

$$\bigwedge_{l=1}^{\binom{|C_p|}{2}} \neg \left[\bigvee_{(i,j)\in\{1,\ldots,q\}^2} \varphi_{(s,r)} \right]$$

many predicates for each cell C_p. Then we must do this for all m cells in Γ. We denote the formula ω_m as the formula that tests if there is an m-colouring of some guard set $\mathcal{G}$. Then

$$\omega_m := \bigwedge_{p=1}^{m} \left[\bigwedge_{l=1}^{\binom{|C_p|}{2}} \neg \left[\bigvee_{(i,j)\in\{1,\ldots,q\}^2} \varphi_{(s,r)} \right] \right]$$

We have now encoded the formulas that check to see if a partition of k many visibility regions $V(g_1), \ldots, V(g_k)$ where k is fixed, is m colourable. The output TRUE will tell us that it is. As was argued in Observation 4.3.1, the chromatic guard number will be less than or equal to k. We encode $S(k,m)$ of these formulas to check if there is an m colouring. When we test for all m that is needed, we must sum over each stirling number $S(k,m)$ for $1 \leq m \leq k$ which gives us the bell number

$$B_k = \sum_{m=0}^{k} S(k,m)$$

We have not needed to encode any more variables than the ones used in Φ, and so we use the exact same ones that they used for that formula. We abbreviate Ω as

$$\Omega := \Phi \wedge \omega_m$$

Size of Ω

For a pair of guards we need to create $O((nk)^2)$ predicates to check if their regions intersect. Then for a given partition $\{C_1, \ldots, C_m\}$ we have $\binom{|C_p|}{2} < \binom{k}{2} = O(k^2)$, and so $O(k^3)$ gives us a bound on the number of pairs of guards we need to check for each partition of size m. So we have at most $O(k^3)O((nk)^2)$=$O(n^2k^5)$ many predicates needed to check that a single partitioned guard set is m-colourable.
We saw that $S(k, m)$ gives us the number of ways of partitioning k guards into m nonempty subsets. A lower bound on $S(k, m)$ is m^{k-m}, which would imply that if $m = k/2$ then this gives us $(k/2)^{(k/2)}$ as a lower bound on checking for a $m = k/2$ colouring of the visibility regions. To find the chromatic guard number, we need to check for all $m \leq k$. So this also gives us a lower bound on the number of predicates needed to encode k many Ω formulas. So the formula length is not polynomially bounded as the exponent grows in the size of the input k.
We are not done yet. Although our formula has fixed size for each k, we havn't said how to describe it. We deal with this by using an algorithm to list all subformulas of the form $\varphi_{(s,r)}$ that we need to encode for, on input of size k. List each of the B_k many partitions we need to encode for. The list tells us the size of the partition, which is m, and the size of each cell $C_1, \ldots, C_m$. From here we generate all of the $\binom{|C_1|}{2} + \cdots + \binom{|C_m|}{2}$ subformulas of the form $\varphi_{(s,r)}$. Attaching this algorithm to Ω implies that the formula is fully described by the input k. We now have a proof of Proposition 4.3.2.

4.3.3 Beyond $\exists\mathbb{R}$

A recent development by Dobbins et al. [29], has been to define (possibly) larger complexity classes $\forall\exists\mathbb{R}$ and $\exists\forall\mathbb{R}$. The definition for these classes is as you'd expect. For real variables $Y = (Y_1, Y_2, \ldots, Y_m)$ and $X = (X_1, X_2, \ldots, X_n)$, then the set of all true sentences $\Phi(X, Y)$ in prenex form

$$(\forall Y = (Y_1, Y_2, \ldots, Y_m))(\exists X = (X_1, X_2, \ldots, X_n)) : \Phi(X, Y)$$

is called the Universal Existential Theory of the Reals (UETR). We define $\forall\exists\mathbb{R}$ as the set of all decision problems that are many to one polynomial time reducible to UETR. The class $\exists\forall\mathbb{R}$ is defined similarly.
In their paper they conjectured that some well known problems in computational geometry were complete in these classes. The containment properties of these classes are

$$\exists\mathbb{R} \subseteq \forall\exists\mathbb{R} \subseteq \text{PSPACE}$$

$$\exists\mathbb{R} \subseteq \exists\forall\mathbb{R} \subseteq \text{PSPACE}$$

It is not known whether these are proper containments or not. It may be that the chromatic art gallery problem is complete for one of these classes.

Chapter 5

Conclusion

5.1 Summary

We have seen that the difficulty of proving NP-membership of various natural geometric problems lead to the introduction of the complexity class $\exists\mathbb{R}$. Many of these geometric problems have been shown to be complete in $\exists\mathbb{R}$, including the original art gallery problem. The difficulty of proving $\exists\mathbb{R}$-completeness for this problem motivated the proof that ETR-INV is $\exists\mathbb{R}$-complete. This is a powerful tool that allowed us to bypass simulating multiplication of real variables when proving $\exists\mathbb{R}$-completeness for a given problem. This tool has already been applied to other $\exists\mathbb{R}$-complete proofs in the literature as seen in [7]. In our discussion on smoothed analysis, we saw that the intractable worst case scenario running times for an $\exists\mathbb{R}$-complete problem can be shown to be rare and lacking a common structure. Finally, we examined the chromatic art gallery problem and proved its decidability by encoding it as a formula in ETR.

5.2 Open problems and future research

There are many open problems and possible directions for future research, not just for visibility but for computational geometry in general. First, it is still an open problem as to which complexity class the chromatic art gallery problem is complete in. Answering this, or achieving a more modest goal of showing it has membership of a complexity class such as the ones described in the last chapter, is an interesting research question. Another direction we could pursue would be to find a low smoothed running time of the chromatic art gallery problem, and to try and develop a practical algorithm that works for typical polygons. This could have some advantages over research that has focused on restricted, and rare types of polygons such as orthogonal polygons [26, 30]. These are polygons where the interior angle of every vertex can only be $\frac{\pi}{2}$ or $\frac{3\pi}{2}$. Even for these cases, finding the chromatic guard number is NP-complete [31].

Outside of visibility theory, there are still many interesting natural geometric problems not yet shown to be complete for a complexity class, and they may well be $\exists\mathbb{R}$-complete. Proving that they are is useful because it gives strong evidence that combinatorial methods, (instead of algebraic) cannot be used to give exact answers to the problem. On the other hand it may also be evidence that they have low smoothed running times, or that typical instances of the problem can be decided using combinatorial methods, as in the case we saw for vision stable polygons. This is speculation though. If more $\exists\mathbb{R}$-complete problems are studied from the perspective of smoothed analysis we could gain further insight into that question.

Bibliography

[1] J. O'Rourke *et al.*, *Art gallery theorems and algorithms*. Oxford University Press Oxford, 1987, vol. 57.

[2] S. K. Ghosh, *Visibility algorithms in the plane*. Cambridge university press, 2007.

[3] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of discrete and computational geometry*. CRC press, 2017.

[4] D. Lee and A. Lin, "Computational complexity of art gallery problems," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 276–282, 1986.

[5] C. McDiarmid and T. Müller, "Integer realizations of disk and segment graphs," *Journal of Combinatorial Theory, Series B*, vol. 103, no. 1, pp. 114–143, 2013.

[6] J. Matousek, "Intersection graphs of segments and $\exists\mathbb{R}$," *arXiv preprint arXiv:1406.2636*, 2014.

[7] N. Bieker, "Complexity of graph drawing problems in relation to the existential theory of the reals," Ph.D. dissertation, Bachelor's thesis, Karlsruhe Institute of Technology (August 2020), 2020.

[8] J. Cardinal and U. Hoffmann, "Recognition and complexity of point visibility graphs," *Discrete & Computational Geometry*, vol. 57, no. 1, pp. 164–178, 2017.

[9] M. Abrahamsen, A. Adamaszek, and T. Miltzow, "Irrational guards are sometimes needed," *arXiv preprint arXiv:1701.05475*, 2017.

[10] M. G. Dobbins, A. Holmsen, and T. Miltzow, "Smoothed analysis of the art gallery problem," *arXiv preprint arXiv:1811.01177*, 2018.

[11] J. O'Rourke *et al.*, *Computational geometry in C*. Cambridge university press, 1998.

[12] S. L. Devadoss and J. O'Rourke, *Discrete and computational geometry*. Princeton University Press, 2011.

[13] D. Mount, "Lectures," 2021. [Online]. Available: https://www.cs.umd.edu/class/fall2021/cmsc754/lectures.html

[14] M. Sipser, "Introduction to the theory of computation," *ACM Sigact News*, vol. 27, no. 1, pp. 27–29, 1996.

[15] S. Fisk, "A short proof of chvátal's watchman theorem," *Journal of Combinatorial Theory, Series B*, vol. 24, no. 3, p. 374, 1978.

[16] R. McNaughton, "Alfred tarski, a decision method for elementary algebra and geometry," *Bulletin of the American Mathematical Society*, vol. 59, no. 1, pp. 91–93, 1953.

[17] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the third annual ACM symposium on Theory of computing*, 1971, pp. 151–158.

[18] J. Canny, "Some algebraic and geometric computations in pspace," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, 1988, pp. 460–467.

[19] M. Abrahamsen, A. Adamaszek, and T. Miltzow, "The art gallery problem is $\exists\mathbb{R}$-complete," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018, pp. 65–73.

[20] A. Efrat and S. Har-Peled, "Guarding galleries and terrains," *Information processing letters*, vol. 100, no. 6, pp. 238–245, 2006.

[21] M. Abrahamsen and T. Miltzow, "Dynamic toolbox for etr inv," *arXiv preprint arXiv:1912.08674*, 2019.

[22] M. Schaefer and D. Štefankovič, "Fixed points, nash equilibria, and the existential theory of the reals," *Theory of Computing Systems*, vol. 60, no. 2, pp. 172–193, 2017.

[23] S. Basu and M.-F. Roy, "Bounding the radii of balls meeting every connected component of semi-algebraic sets," *Journal of Symbolic Computation*, vol. 45, no. 12, pp. 1270–1279, 2010.

[24] S. Hengeveld and T. Miltzow, "A practical algorithm with performance guarantees for the art˜ gallery problem," *arXiv preprint arXiv:2007.06920*, 2020.

[25] L. Erickson and S. M. LaValle, "A chromatic art gallery problem," Tech. Rep., 2010.

[26] A. Bärtschi and S. Suri, "Conflict-free chromatic art gallery coverage," *Algorithmica*, vol. 68, no. 1, pp. 265–283, 2014.

[27] S. P. Fekete, S. Friedrichs, and M. Hemmer, "Complexity of the general chromatic art gallery problem," *arXiv preprint arXiv:1403.2972*, 2014.

[28] N. Megiddo and A. Tamir, "On the complexity of locating linear facilities in the plane," *Operations research letters*, vol. 1, no. 5, pp. 194–197, 1982.

[29] M. G. Dobbins, L. Kleist, T. Miltzow, and P. Rzążewski, "$\forall\exists\mathbb{R}$ -completeness and area-universality," in *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer, 2018, pp. 164–175.

[30] F. Hoffmann, K. Kriegel, S. Suri, K. Verbeek, and M. Willert, "Tight bounds for conflict-free chromatic guarding of orthogonal art galleries," *Computational Geometry*, vol. 73, pp. 24–34, 2018.

[31] H. Hoorfar and A. Bagheri, "Np-completeness of chromatic orthogonal art gallery problem," *The Journal of Supercomputing*, vol. 77, no. 3, pp. 3077–3109, 2021.