

Variations by complexity theorists on three themes of Euler, Bézout, Betti, and Poincaré

Peter Bürgisser and Felipe Cucker

Contents

1. Introduction (3).
2. Topological invariants of (semi)algebraic geometry (9).
3. Models of computation (14).
4. Concrete complexity: some upper bounds (19).
5. Concrete complexity: some lower bounds (26).
6. Structural complexity: basic classes and results (41).
7. Structural complexity: Bézout, Euler, and Betti (53).
8. Open problems (71).

1. Introduction

A first description of what computational complexity is could be something like “the study of the cost of algorithmically solving problems.” While this description is accurate enough it is also very encompassing and is already suggesting the diversity that the subject has developed in the last few decades. In this survey we expect to convey a feeling of this diversity by looking at a number of ways in which a few notions in algebraic geometry and algebraic topology occur in computational complexity.

These notions are the three themes alluded to in the title. We briefly explain what the variations on these themes are at the end of this introduction. Before doing so, however, we precise the meanings of the words “cost”, “algorithmically”, and “problem” (and even that of “study”) in the description above. It is the variety of possible meanings for them which allow for our variations.

1.1. Complexity

1.1.1 A *problem* P is a subset in a product $\mathcal{I} \times \mathcal{O}$, where \mathcal{I} is a set of inputs and \mathcal{O} is a set of outputs, such that for all $x \in \mathcal{I}$ there is $y \in \mathcal{O}$ with $(x, y) \in P$.

A first source of variety is the nature of \mathcal{I} and \mathcal{O} . Consider, for instance, the problem of computing a complex root ξ of a given non-constant univariate polynomial p . Computer algebra practitioners will deal with polynomials p with rational coefficients and will return ξ encoded using rational numbers as well (for instance, giving a small square in the complex plane containing only one root of p). They will probably use `MAPLE` to solve the problem and `MAPLE` will only use rational numbers during the execution. In contrast, numerical analysts will deal with polynomials whose coefficients are floating-point numbers, will probably use `Fortran` to solve the problem, and will return ξ encoded by a pair a floating-point numbers. Floating-point numbers are a finite—but very dense—subset of \mathbb{R} . Their use has the drawback of the unavoidable accumulation of small errors but the advantage of speed. Our numerical analyst will obtain his solution well before our computer algebra practitioner. But

*The first author is partially supported by DFG grant BU 1371; the second author is partially supported by City University SRG grant 7001558.

the only guarantee for the accuracy of this solution is the knowledge that his algorithm is stable and the fact that random polynomials tend to be well-conditioned.¹

A key difference in the way **MAPLE** deals with rational numbers and **Fortran** does so with floating-point numbers is an implicit notion of *size*. A rational number is represented by a pair of integers, and each one of these by a string of bits. The larger (in absolute value) is an integer, the more bits the computer needs to represent it. So, integer and rational numbers have *variable size*. In contrast, all floating-point numbers are represented in a computer using a fixed number of bits. Therefore, they all have the same *fixed size*.

This major division in computing (the one between discrete and continuous data) is simply modelled in complexity theory. Discrete data are modelled by strings in a binary alphabet and floating-point numbers are modelled by real numbers. Therefore, the two main choices for \mathcal{I} considered in complexity theory are the disjoint unions

$$\Sigma^* = \{0, 1\}^\infty = \bigsqcup_{n \geq 0} \{0, 1\}^n$$

for discrete problems and

$$\mathbb{R}^\infty = \bigsqcup_{n \geq 0} \mathbb{R}^n$$

for continuous ones. A third choice, of mainly theoretical relevance, is \mathbb{C}^∞ . For $x \in \mathbb{K}^n$ (here $\mathbb{K} = \{0, 1\}, \mathbb{R}$ or \mathbb{C}) we call n the *size* of x and we denote it by $\text{size}(x)$.

Besides the difference between discrete and continuous problems, other differences can be considered. For instance, the solution $y \in \mathcal{O}$ of an input $x \in \mathcal{I}$ can be unique (e.g., given a square matrix, compute its determinant) or not (e.g., the problem above, where there may be d different outputs for an input p if p has degree d). In what follows, we will restrict our exposition to problems of the first kind only. In addition to the above, the nature of \mathcal{O} determines further distinctions. An important class of problems, called *decision problems*, is that corresponding to $\mathcal{O} = \{0, 1\}$. These problems consist of deciding (i.e.,

¹In this paper we will not deal with the stability issues associated with continuous computations. A survey paper dealing with these issues is [36].

answering) a **yes/no** question (e.g., given a square real matrix A , is A positive definite?) the output 1 corresponding to **yes**. Another important class of problems, called *counting problems*, is that corresponding to $\mathcal{O} = \mathbb{N}$.

1.1.2 Informal definitions of *algorithm* describe the latter as an “unambiguous sequence of instructions.” In complexity theory this is replaced by more formal notions of *machine models* the plural already suggesting that there are many of them. Certainly, the three possible input data mentioned above (i.e., bits, real numbers and complex numbers) induce a first distinction on machine models. To each of these kinds of data, a relevant set of *basic operations* is associated. The relevant operations for $\{0, 1\}$ are the Boolean operations $\{\neg, \vee, \wedge\}$, while for \mathbb{R} the relevant operations are $\{+, -, \times, /\}$ together with the relation \leq , and those for \mathbb{C} are the same but with $=$ instead of \leq . Machine models are usually defined by associating to these operations some form of data management and some output convention.

A notion of *cost* is normally associated with the basic operations (and to the data management operations such as copying data from one register to another, or moving a reading head one cell to the left or the right). Usually, it is assumed that the cost of such an operation is 1. This induces a natural notion of cost for the computation of a machine on an input $x \in \mathcal{I}$, denoted by $\text{cost}(x)$.²

To assess how efficiently a problem can be solved, the values of $\text{cost}(x)$ for a few individual $x \in \mathcal{I}$ is of little relevance. Instead, complexity theorists wrap up the behavior of a machine in a function whose asymptotic growth describe the cost of solving increasingly large inputs. The two main such functions are the *worst-case cost function*

$$\text{worst} : \mathbb{N} \rightarrow \mathbb{N}, n \mapsto \sup_{x \in \mathcal{I}_n} \text{cost}(x)$$

and the *average-case cost function*

$$\text{avg} : \mathbb{N} \rightarrow \mathbb{R}_+, n \mapsto \mathbf{E}_{x \in \mathcal{I}_n}^{\mu_n} \text{cost}(x).$$

²This notion of cost is related to *time*. Other resources, notably space, can also induce cost notions, but we will not deal with them in this paper.

Here \mathcal{I}_n is the set of inputs of size n and \mathbf{E}^{μ_n} denotes the expected value with respect to a given probability measure μ_n in \mathcal{I}_n . While the average case is considered a more “realistic” measure of algorithmic efficiency it has the drawback of depending on a probability measure whose choice is arbitrary.

To these cost functions one associates the expression “the complexity (or cost) of an algorithm.” If \mathcal{A} is an algorithm solving a problem P we say that \mathcal{A} has complexity f when its worst-case cost function is bounded by f . Assertions like “ \mathcal{A} has polynomial complexity” or “ \mathcal{A} has quadratic complexity” have obvious meanings and the same holds for assertions involving “the average complexity of \mathcal{A} .”

1.1.3 We have just seen the meanings of the words “cost”, “algorithmically”, and “problem” in the description of computational complexity as “the study of the cost of algorithmically solving problems.” We finally come to the meaning of “study.”

The ultimate goal of complexity theory is to unveil the inherent complexity of computational problems. That is, given a problem P , to find matching upper and lower bounds for its complexity. But, again, this can be done at several levels.

At the most concrete level, an upper bound for the complexity of P is a function f such that there exists an algorithm solving P whose cost function is bounded above by f (here the bound can be on the worst-case or average-case complexity of P depending on the cost function we are bounding). Also, a lower bound for the complexity of P is a function f such that all algorithms solving P , within a certain machine model, have a cost function bounded below by f . This concrete level can deal with very fine distinctions. For instance, it is known that the average-case complexity of sorting (given $x \in \mathbb{R}^n$, permute its components such that, after permutation, $x_i \leq x_j$ when $i < j$) is $\Omega(n \log n)$. This lower bound is optimal since several sorting algorithms have average-cost (or even worst-case cost) $\mathcal{O}(n \log n)$. A good amount of effort was further devoted to study the constants implicit in the \mathcal{O} notation to select the fastest one.

At a more abstract level, complexity theory clusters problems in *complexity classes*. These classes are then related by a large number of relations (notably

the inclusion) which draw a landscape of conceptual levels of difficulty for computational problems. Complexity classes are defined by fixing a machine model and a family of bounds (usually invariant under polynomial maps, e.g., polynomial functions, exponential functions, etc.).³ The classes P and NP of the decision problems which can be solved by deterministic and nondeterministic machines in polynomial time, respectively, are the best known examples. At this level of abstraction, an upper bound for a problem P is the proof of membership of P to a class \mathcal{C} . A lower bound for P is the proof that P is more difficult (in a sense to be precised in §6) than all problems in \mathcal{C} . Such problems are said to be \mathcal{C} -hard. A \mathcal{C} -hard problem belonging to \mathcal{C} is said to be \mathcal{C} -complete. Thus a proof of completeness in a class is an optimality result in the sense of matching upper and lower bounds.

Throughout this paper, we will use the words *concrete* and *structural* to distinguish between the two kinds of bounds above.

1.2. Euler, Bézout, Betti and Poincaré

1.2.1 The study of the zero sets of systems of polynomial equations is the subject of algebraic geometry. Classically, these zero sets, called algebraic varieties, are considered in K^n for some algebraically closed field K . A central choice is $K = \mathbb{C}$. Given an algebraic variety Z , a number of quantities are attached to it, which describe several geometric features of Z . Examples of such quantities are dimension and degree. Roughly speaking, the degree measures how twisted Z is embedded in affine space by, more precisely, counting how many intersection points it has with generic affine subspaces of a certain well-chosen dimension. Not surprisingly, an algebraic variety has degree one if and only if it is an affine subspace of \mathbb{C}^n . The degree of an algebraic variety occurs in many results in algebraic geometry. Maybe the most celebrated of them is Bézout's Theorem. It also occurs in the algorithmics of algebraic geometry [44, 60] and in lower bounds results [25, 109].

³This polynomial invariance makes the class robust to two sorts of events: a change in the problem encoding (affecting input size by no more than a polynomial factor) and minor variations in the machine model (such as considering multitape Turing machines instead of single tape Turing machines).

1.2.2 The birth of algebraic topology is entangled with more than one century of attempts to prove a statement of Euler asserting that in a polyhedron the number of vertices plus the number of faces minus the number of edges equals 2 (see [79] for a vivid account of this history). A precise definition of a generalization of this sum is today (justly) known with the name of Euler characteristic or (justly as well) of Euler-Poincaré characteristic.

The Euler characteristic of X , denoted by $\chi(X)$, is one of the most basic invariants in algebraic topology. Remarkably, it naturally occurs in many applications in other branches of geometry. For instance, in differential geometry, where it is proved that a compact, connected, differentiable manifold X has a non-vanishing vector field if and only if $\chi(X) = 0$ [108, p. 201]. Also, in algebraic geometry, a generalization of the Euler characteristic (w.r.t. sheaf cohomology) plays a key role in the Riemann-Roch Theorem for non-singular projective varieties [65]. The Euler characteristic has also played a role in complexity lower bounds results. For this purpose, Yao [118] introduced a minor variation of the Euler characteristic. This *modified Euler characteristic* has a desirable additivity property and coincides with the usual Euler characteristic in many cases, e.g., for compact semialgebraic sets and complex algebraic varieties.

1.2.3 A goal in algebraic topology is to classify topological spaces. To this end, one attaches to these spaces a number of objects whose invariance under different notions (e.g., homeomorphism, homotopy equivalence, etc.) helps to distinguish non-equivalent objects under these notions. For instance, the Euler characteristic is invariant under homotopy equivalence. And we know that $\chi(S^2) = 2$ and $\chi(T^2) = 0$ where S^2 and T^2 are the 2-dimensional sphere and torus respectively. Therefore, we conclude that the surface of an orange and that of a doughnut are not homotopically equivalent. The Euler characteristic, however, fails to distinguish non-equivalent spaces. For instance, both S^1 and S^3 (the 1-dimensional and 3-dimensional spheres respectively) have Euler characteristic 0. A more powerful object is the sequence of *Betti numbers*. This is a sequence of non-negative integers $b_k(X)$ associated to a topological space X , which satisfies $b_k(X) = 0$ for all k strictly larger than the dimension of X . In addition, $b_0(X)$ has a very simple meaning: it is the number of connected

components of X . Thus, $b_0(S^1) = b_0(S^3) = 1$, but in addition $b_1(S^1) = 1$, $b_1(S^3) = b_2(S^3) = 0$, and $b_3(S^3) = 1$. This shows that S^1 and S^3 are not homotopically equivalent (as one could well expect).

The degree, Euler characteristic, and Betti numbers appear in this paper in two roles. On the one hand, they are used to prove concrete lower bounds for several continuous problems. This role will be featured in §5. On the other hand, the computation of these quantities on a number of sets (complex varieties, semialgebraic sets) yields a number of corresponding counting problems whose complexity, both concrete and structural, is of interest and becomes the subject of §7.

To do the above, a more formal description of the concepts informally described before is needed. We devote §2 to such a description of degree, Euler characteristic and Betti numbers as well as to the sets for which these quantities will be considered in this paper. Also, §3 does so for several machine models.

2. Topological invariants of (semi)algebraic geometry

2.1. Algebraic and semialgebraic sets

Algebraic geometry is the study of zero sets of polynomials (or of objects which locally resemble these sets). These zero sets are usually considered in \mathbb{C}^n (or, more generally, in K^n for some algebraically closed field K).

We very briefly recall some definitions and facts from algebraic geometry, which will be needed later on. Standard textbooks on algebraic geometry, where detailed treatments of the subject can be found, are [58, 91, 105].

An *algebraic set* (or *affine algebraic variety*) Z is defined as the zero set

$$Z = \mathcal{Z}(f_1, \dots, f_r) := \{x \in \mathbb{C}^n \mid f_1(x) = 0, \dots, f_r(x) = 0\}$$

of finitely many polynomials $f_1, \dots, f_r \in \mathbb{C}[X_1, \dots, X_n]$. The *vanishing ideal* $\mathcal{I}(Z)$ of Z consists of all the polynomials vanishing on Z . Note that $\mathcal{I}(Z)$ might be strictly larger than the ideal I generated by f_1, \dots, f_r . Actually, by

Hilbert's Nullstellensatz, $\mathcal{I}(Z)$ can be characterized as the so-called radical of the ideal I ,

$$\mathcal{I}(Z) = \sqrt{I} := \{f \in \mathbb{C}[X_1, \dots, X_n] \mid f^q \in I \text{ for some } q \in \mathbb{N}\}.$$

A usual compactification of the space \mathbb{C}^n consists of embedding \mathbb{C}^n into $\mathbb{P}^n(\mathbb{C})$, the *projective space* of dimension n over \mathbb{C} . This is the set of complex lines through the origin in \mathbb{C}^{n+1} . Note that \mathbb{C}^n can be embedded in $\mathbb{P}^n(\mathbb{C})$ by

$$\mathbb{C}^n \hookrightarrow \mathbb{P}^n(\mathbb{C}), \quad x \mapsto \ell(x),$$

where $\ell(x)$ is the line in \mathbb{C}^{n+1} passing through the origin and through $(1, x)$. The notion of an affine algebraic variety extends to that of a *projective variety* by replacing polynomials by homogeneous polynomials in $\mathbb{C}[X_0, X_1, \dots, X_n]$, for which elements of $\mathbb{P}^n(\mathbb{C})$ are natural zeros. The embedding above extends to the algebraic subsets of \mathbb{C}^n by defining, for any such set Z , its *projective closure* \bar{Z} as the smallest projective variety in $\mathbb{P}^n(\mathbb{C})$ containing Z .

Zero sets of real polynomials in \mathbb{R}^n are, needless to say, also of interest. It turns out, however, that the proper frame to study these sets is a more general one in which the object of study are sets defined not only by equalities but also by inequalities on real polynomials. The resulting subject is called real algebraic geometry (or semialgebraic geometry). Again, we next briefly recall some definitions and facts of this subject. For detailed expositions we refer to [11, 18].

A *basic semialgebraic set* $S \subseteq \mathbb{R}^n$ is defined to be a set of the form

$$(2.1) \quad S = \{x \in \mathbb{R}^n \mid g(x) = 0, f_1(x) > 0, \dots, f_r(x) > 0\},$$

where g, f_1, \dots, f_r are polynomials in $\mathbb{R}[X_1, \dots, X_n]$. We say that $S \subseteq \mathbb{R}^n$ is a *semialgebraic set* when it is a Boolean combination of basic semialgebraic sets in \mathbb{R}^n . Every semialgebraic set S can be represented as a finite union $S = S_1 \cup \dots \cup S_t$ of basic semialgebraic sets. If we require that the degrees of the polynomials in the description are at most one, then the resulting set S is called *semilinear*.

We next have a closer look at our three themes.

2.2. Notions of degree

The degree of an algebraic variety Z embedded in affine or projective space can be seen as a measure for the degree of nonlinearity of Z . A detailed treatment of this notion can be found in standard textbooks on algebraic geometry [58, 91, 105].

We will consider two different notions of degree.

Definition 2.1.

- (i) *The degree $\deg Z$ of an irreducible affine algebraic set $Z \subseteq \mathbb{C}^n$ (or irreducible projective algebraic set $Z \subseteq \mathbb{P}^n(\mathbb{C})$) of dimension d is the number of intersection points of Z with a generic affine (or linear) subspace of codimension d .*
- (ii) *The (geometric) degree $\text{DEG } Z$ of a reducible algebraic set Z is the sum of the degrees of the irreducible components of Z of maximal dimension.*
- (iii) *The cumulative degree $\deg Z$ of a reducible algebraic set Z is the sum of the degrees of all the irreducible components of Z .*

The geometric degree is the usual notion of degree studied in algebraic geometry. Its characterization given in (i) for irreducible algebraic sets does also hold for reducible algebraic sets. However the cumulative degree, introduced by Heintz [60], turns out to be more useful for complexity estimates, as we will see shortly. Clearly, $\deg Z = \text{DEG } Z$ for irreducible algebraic sets. Note that $\deg \emptyset = \text{DEG } \emptyset = 0$.

Bézout's theorem relates the degree of the intersection of two varieties with the degrees of the varieties themselves. In algebraic complexity theory, the Bézout Inequality 2.1 given below is a fundamental tool. It can be derived in a straightforward way from the version of Bézout's theorem treating the intersection of an irreducible variety with an irreducible hypersurface, see for instance [58, Thm. I.7.7] and [25].

Bézout Inequality 2.1. *Let Z be an affine algebraic variety in \mathbb{C}^n and H be a hypersurface. Then we have $\deg(Z \cap H) \leq \deg Z \cdot \deg H$.*

Remark 2.1 - The Bézout Inequality does not hold for DEG instead of deg. As a counterexample take for Z the union of a plane Z_1 with a line Z_2 in \mathbb{C}^3 and let H be a plane containing Z_2 and intersecting Z_1 properly. Then $\text{DEG } Z = 1$, $\text{deg } Z = 2$ and $\text{DEG}(Z \cap H) = \text{deg}(Z \cap H) = 2$. This is this reason why the cumulative degree is more useful for complexity estimates. Bézout's Inequality 2.1 does also hold for the intersection of any two locally closed subsets [60] (see also [25, p. 201]).

Remark 2.2 - One can assign to an irreducible projective variety Z of dimension d a homology class $[Z] \in H_{2d}(Z; \mathbb{Z})$ in a natural way (if Z is smooth, then $[Z]$ is the fundamental class of Z considered as a compact oriented manifold Z of dimension $2d$). If i denotes the embedding $i: Z \rightarrow \mathbb{P}^n(\mathbb{C})$ then we have $i_*([Z]) = \text{deg } Z \cdot [L]$, where $[L] \in H_{2d}(\mathbb{P}^n(\mathbb{C}); \mathbb{Z})$ is the homology class of a d -dimensional linear subspace of $\mathbb{P}^n(\mathbb{C})$ (cf. [57, p. 226]). This shows that the degree is a purely topological property of the embedding of Z in $\mathbb{P}^n(\mathbb{C})$.

2.3. Euler characteristic

For the remaining two themes we consider semialgebraic sets.

It is well known that any compact semialgebraic set S can be triangulated [18, § 9.2]. Instead of working with triangulations, we will use the more general notion of finite cell complexes. Compact semialgebraic sets are homeomorphic to finite cell complexes and their topology can be studied through the combinatorics of cell complexes.

We briefly recall the definition of a finite cell complex (also called finite CW-complex), see, for instance, [59] for more details. We denote by D^n the closed unit ball in \mathbb{R}^n , and by $S^{n-1} = \partial D^n$ its boundary, the $(n-1)$ -dimensional unit sphere. An n -disk is a space homeomorphic to D^n . By an *open n -cell* we understand a space e^n homeomorphic to the open unit ball $D^n - \partial D^n$. A (finite) *cell complex* X is obtained by the following inductive procedure.

We start with a finite discrete set X^0 , whose points are regarded as 0-cells. Inductively, we form the n -skeleton X^n from X^{n-1} by attaching a finite number of open n -cells e_α^n via continuous maps $\varphi_\alpha: S^{n-1} \rightarrow X^{n-1}$. This means that X^n is the quotient space of the disjoint union $X^{n-1} \sqcup_\alpha D_\alpha^n$ of X^{n-1} with a finite collection of n -disks D_α^n under the identifications $x \equiv \varphi_\alpha(x)$ for

$x \in \partial D_\alpha^n = S^{n-1}$. Thus as a set, $X^n = X^{n-1} \sqcup_\alpha e_\alpha^n$, where each e_α^n is an open n -cell. We stop this procedure after finitely many steps obtaining the compact space $X = X^d$ of dimension d .

Example 2.1 -

- (i) The n -sphere S^n can be realized as a cell complex with two cells, of dimension 0 and n , respectively. The cell e^n is attached to e^0 by the constant map $\varphi : S^{n-1} \rightarrow e^0$.
- (ii) Real projective space $\mathbb{P}^n(\mathbb{R})$ is defined as the space of all lines through the origin in \mathbb{R}^{n+1} . This is equivalent to identify antipodal points in $S^n \subset \mathbb{R}^{n+1}$, a presentation which in addition yields a natural topology in $\mathbb{P}^n(\mathbb{R})$ —the quotient topology induced by the identification. Removing the southern hemisphere, this is yet equivalent to the space obtained by keeping the northern hemisphere and identifying antipodal points in the equator. Since the northern hemisphere (without the equator) is homeomorphic to e^n and the equator with identified antipodal points is just $\mathbb{P}^{n-1}(\mathbb{R})$, it follows that $\mathbb{P}^n(\mathbb{R})$ is obtained from the $n + 1$ cells e^0, e^1, \dots, e^n by taking $X^0 = e^0$ and, inductively, obtaining $X^k = \mathbb{P}^k(\mathbb{R})$ from X^{k-1} by attaching e^k via the identification of antipodal points $\varphi_k : \partial D^k \rightarrow X^{k-1}$.

The *Euler characteristic* of a cell complex X is defined as $\chi(X) = \sum_{k=0}^d (-1)^k N_k$, where N_k is the number of k -cells of the complex. It is a well-known fact that $\chi(X)$ depends only on the topological space X and not on the cellular decomposition. That is, if two cell complexes are homeomorphic, then their Euler characteristics are the same. Actually, χ is even a homotopy invariant.

Example 2.1 - (continued) For the spaces considered above we obtain, using their cell decompositions, that $\chi(S^n) = 1 + (-1)^n$, $\chi(\mathbb{P}^n(\mathbb{R})) = \sum_{k=0}^n (-1)^k$, thus

$$\chi(S^n) = \begin{cases} 2 & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases} \qquad \chi(\mathbb{P}^n(\mathbb{R})) = \begin{cases} 1 & \text{if } n \text{ is even} \\ 0 & \text{if } n \text{ is odd} \end{cases} .$$

2.4. Betti numbers

There are several ways to extend the definition of χ to non-compact sets. The usual one is using singular homology, which preserves the property of χ of being a homotopy invariant. In §5.3 we will see another way which does not, but instead has a useful additivity property.

In algebraic topology one assigns to a topological space X and a field F the singular *homology vector spaces* $H_k(X; F)$ for $k \in \mathbb{N}$, which depend only on the homotopy type of X and on F . The *kth Betti number* of X with respect to F , denoted $b_k(X; F)$, is defined as the dimension of $H_k(X; F)$. In case $F = \mathbb{Q}$ we write $b_k(X)$ and talk about the *kth Betti number* of X . The Euler characteristic of the space X is defined by

$$(2.2) \quad \chi(X) = \sum_{k \in \mathbb{N}} (-1)^k \dim_F H_k(X; F)$$

(if this sum is finite). The Betti numbers $b_k(X; F)$ depend on the field F as well as on X . Remarkably, their alternate sum is independent of F . In addition, for cell complexes X , this alternate sum coincides with $\chi(X)$ as defined in §2.3. For a general reference to homology we refer to [59, 92].

More generally, one can assign to a pair $Y \subseteq X$ of topological spaces the *relative Euler characteristic* $\chi(X, Y) := \chi(X) - \chi(Y)$. It can also be characterized in terms of the *relative homology vector spaces* $H_k(X, Y; F)$ as $\chi(X, Y) = \sum_{k \in \mathbb{N}} (-1)^k \dim_F H_k(X, Y; F)$. Since $H_k(X, Y; F)$ depends only on the homotopy type of the pair (X, Y) , the same holds for the relative Euler characteristic $\chi(X, Y)$. Note that $H_k(X, \emptyset; F) = H_k(X; F)$ and $\chi(X, \emptyset) = \chi(X)$.

3. Models of computation

In this section we describe a few models of computation giving formal substance to the notion of “algorithm.” These descriptions are short, but we will point to adequate references for detailed accounts.

3.1. Algebraic circuits

Definition 3.1. An algebraic circuit \mathcal{C} over \mathbb{R} is an acyclic directed graph where each node has indegree 0, 1, or 2. Nodes with indegree 0 are either

labeled as input nodes or with elements of \mathbb{R} (we shall call these constant nodes). Nodes with indegree 2 are labeled with one of $\{+, -, \times, /\}$. They are called arithmetic nodes. Nodes with indegree 1 are output nodes or sign nodes. All output nodes have outdegree 0. Otherwise, there is no upper bound on the outdegree of the other nodes. For an algebraic circuit \mathcal{C} , the size of \mathcal{C} is the number of nodes in \mathcal{C} . The depth of \mathcal{C} is the length of the longest path from some input node to some output node.

A sign node, with input $x \in \mathbb{R}$, returns 1 if $x \geq 0$ and 0 otherwise. The semantics of all other nodes is obvious. If \mathcal{C} is an algebraic circuit with n input nodes and m output nodes, we may talk about the function $\varphi_{\mathcal{C}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ computed by the circuit. (We can make sure that no divisions by zero occur by introducing additional sign nodes.)

A *straight-line program* is an algebraic circuit without sign nodes together with a numbering of the nodes such that successor nodes get higher numbers. It is said to be *division-free* if it has no nodes labeled with $'/'$.

A circuit is said to be *decisional* when it has only one output node which, in addition, returns an element in $\{0, 1\}$ (This can be ensured, for instance if the node preceding the output node is a sign node.) Note that, if \mathcal{C} is a decision circuit, then the set $S_{\mathcal{C}} := \{x \in \mathbb{R}^n \mid \varphi_{\mathcal{C}}(x) = 1\}$ is semialgebraic. Conversely, for every semialgebraic set S there is a circuit deciding S , i.e., such that $S_{\mathcal{C}} = S$. This motivates the question, which are the smallest size or depth of algebraic circuits deciding a semialgebraic set S ? We define the *decision complexity* $C(S)$ and the *parallel decision complexity* $D(S)$ of a semialgebraic set S as follows:

$$C(S) := \min\{\text{size}(\mathcal{C}) \mid S_{\mathcal{C}} = S\}, \quad D(S) := \min\{\text{depth}(\mathcal{C}) \mid S_{\mathcal{C}} = S\}.$$

Remark 3.1 - Versions over \mathbb{C} of the above are defined in the obvious way. Versions over $\{0, 1\}$, known as *Boolean circuits*, are defined by taking Boolean nodes (labeled with $\{\neg, \vee, \wedge\}$, the first with indegree one) instead of arithmetic nodes. In this case, there are no sign nodes and consequently, no distinction between Boolean circuits and Boolean straight-line programs is made.

Remark 3.2 - The model of arithmetic networks from [48] distinguishes between the real and Boolean data types. Besides arithmetic and Boolean nodes,

there are sign and selector nodes forming an interface between the different data types. One can show that algebraic circuits and arithmetic networks can simulate each other with respect to both size and depth within a constant factor.

3.2. Blum-Shub-Smale model

The machine models considered so far are finite dimensional. That is, their inputs belong to \mathbb{K}^n for some fixed n (here $\mathbb{K} = \mathbb{R}, \mathbb{C}$ or $\{0, 1\}$). While problems solved by these machines are of interest, the focus of complexity theory is on problems with arbitrarily large inputs. The question of how much does it cost to compute the determinant of a square matrix is given more attention than the same question for, say, 100×100 matrices.

Recall that \mathbb{K}^∞ is the disjoint union $\bigsqcup_{n \geq 0} \mathbb{K}^n$. The space \mathbb{K}^∞ is a natural one to represent problem instances of arbitrarily large size and thus serves as the input space \mathcal{I} for infinite-dimensional machines.

A way to build infinite-dimensional machines from finite dimensional ones is to consider families $\{M_i\}_{i \in \mathbb{N}}$ of machines, each M_i taking inputs in \mathbb{K}^i . This procedure has two drawbacks. Firstly, it does not impose any relationship between the computations for different input sizes. Secondly, it is against the idea of a machine as a finite list of instructions. These machine models are called *non-uniform* and are mainly studied in connection with lower bound results. In contrast, *uniform* models not suffering from the drawbacks above have more practical importance.

The most well-known uniform machine model over $\{0, 1\}$ is the Turing machine. Versions of this model over \mathbb{R} and \mathbb{C} were introduced by Blum, Shub and Smale [17] and are known as BSS-machines. Roughly speaking, such a machine takes an input from \mathbb{R}^∞ (or \mathbb{C}^∞), performs a number of arithmetic operations and comparisons following a finite list of instructions, and either halts returning an element in \mathbb{R}^∞ (resp. \mathbb{C}^∞) or loops forever. For details see [16, 17].

For a given machine M , the function φ_M associating its output to a given input $x \in \mathbb{K}^\infty$ is called the *input-output function*. We shall say that a function $f: \mathbb{K}^\infty \rightarrow \mathbb{K}^q$, $q \leq \infty$, is *computable* when there is a machine M such that $f = \varphi_M$. Also, a set $A \subseteq \mathbb{K}^\infty$ is *decided* by a machine M if its characteristic

function $1_A: \mathbb{K}^\infty \rightarrow \{0, 1\}$ coincides with φ_M . So, for decision problems we consider machines whose output space \mathcal{O} is $\{0, 1\}$.

A simple model for parallel computation consists of considering families of circuits. Let $f: \mathbb{K}^\infty \rightarrow \mathbb{K}^\infty$. The family of algebraic circuits $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ computes f if for all $n \geq 1$, $\varphi_{\mathcal{C}_n}$ is the restriction of f to \mathbb{K}^n . If, in addition, f is the characteristic function of $A \subseteq \mathbb{K}^\infty$ then we say that $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ decides A .

To make a family $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ a uniform model of computation we need to force some relationship between the elements of the family. To fix ideas, we next describe how this is done when $\mathbb{K} = \mathbb{R}$ (over $\{0, 1\}$ or \mathbb{C} a similar idea applies). Now note that a node of an algebraic circuit over \mathbb{R} can be described by four real numbers, say. Thus, a circuit \mathcal{C} with N nodes can be described by a point in \mathbb{R}^{4N} .

Definition 3.2. A family of circuits $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ is said to be uniform if there exists a machine M that, on input (n, i) , outputs the description of the i th node of \mathcal{C}_n . If M works in time $n^{O(1)}$, we shall say that the family is $P_{\mathbb{R}}$ -uniform.

Let $p: \mathbb{N} \rightarrow \mathbb{N}$, $p(n) \geq n$, and $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ be a $P_{\mathbb{R}}$ -uniform family of circuits computing a function f such that \mathcal{C}_n has depth at most $p(n)$. In this case we say that f is computed in parallel time $p(n)$.

Remark 3.3 - One can alternatively use Turing machines to define a notion of P -uniformity for families of circuits over \mathbb{R} or \mathbb{C} (see §6.1 for more details).

3.3. Computation trees

Algebraic computation trees are a somewhat unrealistic, but very powerful model of computation, that is mainly used for proving lower complexity bounds.

Definition 3.3. By an algebraic computation tree T over \mathbb{R} with input variables x_1, \dots, x_n we understand a tree with three types of nodes: arithmetic nodes (outdegree 1), sign nodes (outdegree 3), and leaves nodes (outdegree 0). We assign to each node v of T a variable y_v . To each arithmetic node v there is associated an arithmetic operation $y_v = a \circ b$, where $\circ \in \{+, -, \times, /\}$ and a, b are either real constants, input variables, or variables associated to predecessor nodes of v . To each sign node v , there is associated a real constant, an

input variable, or a variable associated to a predecessor node of v . To each leaf there is associated an output instruction consisting of a list of constants, input variables, or variables associated to predecessor nodes of v . Moreover, each leaf carries a label **yes** or **no**.

The semantics of an algebraic computation tree T is informally described as follows. On an input $x \in \mathbb{R}^n$, the arithmetic operations and sign tests are successively executed (the three successors of a sign node v correspond to the three possible signs of y_v). Accordingly, the computation follows a unique path in the tree from the root to a leaf v (we assume that no attempt to divide by zero is made). The set D_v consisting of all inputs $x \in \mathbb{R}^n$ whose path ends up in the leaf v will be called the *leaf set* of v . Clearly, the (nonempty) leaf sets form a partition of \mathbb{R}^n . The set S_T of inputs x , whose path ends up with a **yes** leaf, is called the set accepted by the tree T . Thus S_T is the disjoint union of the leaf sets D_v corresponding to **yes** leaves. We also say that T solves the membership problem for S_T . The set S_T is clearly a semialgebraic set in \mathbb{R}^n , and all semialgebraic subsets of \mathbb{R}^n may be obtained this way. Note that if all output instructions have the same length m , then T computes a function $\varphi_T: \mathbb{R}^n \rightarrow \mathbb{R}^m$ in an obvious way.

When studying decisional problems, we may assume without loss of generality, that there are no divisions, since a quotient p/q can be encoded by the pair (p, q) , and an arithmetic operation on such a quotient can be simulated by at most 4 arithmetic operations on the numerators and denominators. In the sequel we will therefore always assume that there are no divisions. We remark that sometimes instead of ternary, binary trees are considered (branching on $y_v \geq 0$). More details can be found in [16, 25].

We may interpret algebraic circuits and algebraic computation trees as two different ways of representing semialgebraic sets. An algebraic circuit of size s can be simulated by an algebraic computation tree of depth s . Indeed, after fixing the sequential order of execution in a circuit, the corresponding tree is nothing but an “acyclic flow chart” for the computation. Thus, a lower bound on the depth of algebraic computation trees accepting a semialgebraic set S is in particular a lower bound on the decision complexity $C(S)$, which is defined using algebraic circuits. All the lower bounds on $C(S)$ proven in

§5 are actually lower bounds on the depth of algebraic computation trees. For a discussion of relationships between the models of algebraic circuits and algebraic computation trees we refer to [76].

Remark 3.4 -

- (i) A more restricted model is that of algebraic decision trees. These are ternary trees whose inner nodes v are labeled by polynomials f_v . On input x , the computation starts at the root and branches at node v according to the sign of $f_v(x)$. Again the leaves carry a **yes** or **no**-label. In this model, we count only the number of branchings, but we put a restriction on the f_v by requiring that their degree is less than or equal to some a priori bound d . The model of linear decision trees is obtained by the restriction $d = 1$. Usually, lower bounds are considerably easier to prove in these restricted models.
- (ii) The Boolean version of the decision tree is too powerful. Any subset of $\{0, 1\}^n$ could be decided by such a tree in depth n .
- (iii) Meyer auf der Heide's [85] result providing linear decision trees of size $n^{\mathcal{O}(1)}$ for deciding the real knapsack problem may also be interpreted in the sense that the computation tree model over the reals is too powerful.

4. Concrete complexity: some upper bounds

In this section we list some known upper bounds for a variety of problems. In §4.1 these bounds are low degree polynomials. Thus, the problems considered can be solved very efficiently (with the exception of the real knapsack for which the upper bound is non-uniform and in the unrealistic computation tree model). By contrast, the problem of quantifier elimination dealt with in §4.2 is of a very general nature and the upper bounds exhibited are exponential. Finally, in §4.3, we briefly review the state of the art for computing some topological invariants of semialgebraic sets, without attempting to always present the best known upper bounds. The invariants considered are: dimension, cardinality (of zero dimensional sets), number of connected components, Euler characteristic, and

Betti numbers. We also mention some results about computing invariants of complex algebraic varieties.

4.1. Some polynomial upper bounds

We list here a couple of fundamental computational or decisional problems over the real numbers and indicate a rapid algorithmical solution for each of these problems. In the complexity bounds mentioned, the symbol $M(n)$ will stand for an upper bound on the complexity to multiply two univariate polynomials of degree n over \mathbb{R} . We have $M(n) = \mathcal{O}(n \log n)$, when counting all arithmetic operations, and $M(n) = \mathcal{O}(n)$, when only the nonscalar multiplications and divisions are counted [25].

1. *Element Distinctness*. One has to decide for given real numbers x_1, \dots, x_n , whether they are pairwise distinct. Note that this means to test membership to the complement of a certain arrangement of real hyperplanes. It can be solved by sorting the given real numbers with $\mathcal{O}(n \log n)$ comparisons. An algebraic solution is to first compute the discriminant $\prod_{i < j} (x_i - x_j)^2$ with $\mathcal{O}(M(n) \log n)$ arithmetic operations [9] and then testing the result for zero.

2. *k-equal Problem*. Given n real numbers, decide whether k of them are equal. Using a sorting argument, one can show that $\mathcal{O}(n \log \frac{n}{k})$ comparisons are sufficient to solve this problem (in the linear decision tree model) [13, 14].

3. *Convex Hull*. We want to compute the convex hull of n given points in the real plane. The output of the algorithm should consist of those input points, which are extremal points of the convex hull. A related decision problem is to find out, whether all the input points are extremal. Basic algorithms in computational geometry show that this can be done with $\mathcal{O}(n \log n)$ arithmetic operations and tests (cf. [98]).

4. *Root Verification*. Given n real numbers x_1, \dots, x_n and a real polynomial $p(t) = \sum_{i=0}^n (-1)^i y_i t^{n-i}$ (by its coefficients $y_0 = 1, y_1, \dots, y_n$), verify whether the collection of the x_i forms the complete set of roots of $p(t)$ (including multiplicities). If we denote by σ_i the i th elementary symmetric polynomial in n variables, then this question amounts to test whether $y_i = \sigma_i(x_1, \dots, x_n)$ for all i . There are algorithms which, given $x \in \mathbb{R}^n$, evaluate on x all the elementary symmetric polynomials in n variables using $\mathcal{O}(M(n) \log n)$ arithmetic

operations [109].

5. *Real Knapsack*. This is a membership problem to the union of a certain arrangement of real hyperplanes. One has to decide for given real numbers x_1, \dots, x_n whether there is a set of indices $I \subseteq \{1, \dots, n\}$ such that $\sum_{i \in I} x_i = 1$. The corresponding problem over the rationals is NP-complete [67]. Therefore, it is quite astonishing that the Real Knapsack Problem can be solved by linear decision trees with $\mathcal{O}(n^4 \log n)$ tests, as shown by Meyer auf der Heide [85] (extensions to arbitrary hyperplane arrangements have been obtained by Meiser [84]). It seems that the computation tree model is too powerful to capture the “true” complexity of this problem.

6. *Continued Fraction and GCD*: Let A_1 and A_2 be real univariate polynomials with degrees $n = \deg A_1 \geq \deg A_2$ given by their coefficients. We are required to compute the (coefficients of the) quotients Q_1, \dots, Q_{t-1} occurring in the Euclidean algorithm

$$A_i = Q_i A_{i+1} + A_{i+2}, \deg A_{i+2} < \deg A_{i+1}, A_{i+1} \neq 0, 1 \leq i \leq t-1, A_{t+1} = 0,$$

as well as the greatest common divisor A_t of A_1 and A_2 . Note that the sequence (Q_1, \dots, Q_{t-1}) constitutes the continued fraction expansion of A_1/A_2 . By the *Euclidean representation* of the pair (A_1, A_2) we understand the extension of this sequence by the gcd A_t and we call

$$d := (d_1, \dots, d_{t-1}, d_t) := (\deg Q_1, \dots, \deg Q_{t-1}, \deg A_t)$$

the corresponding degree pattern.

We remark that by Sturm’s theorem (cf. [18]) one can compute the number of zeros of a squarefree polynomial A_1 in any real interval from the Euclidean representation of A_1 and its derivative $A_2 = A_1'$ using $\mathcal{O}(n)$ arithmetic operations. For more information about the Euclidean algorithm and its countless applications, we refer to [49].

The Knuth-Schönhage algorithm [68, 80, 88, 103] is a sophisticated efficient algorithm for computing the Euclidean representation of given polynomials A_1, A_2 with $\mathcal{O}(M(n) \log n)$ arithmetic operations. A detailed analysis of this algorithm was done by Strassen [110]. Let $H(d) := -\sum_{d_i > 0} \frac{d_i}{n} \log \frac{d_i}{n}$ denote the entropy of the degree pattern d . Then the Knuth-Schönhage algorithm

actually uses $\mathcal{O}(n(1 + H(d)))$ arithmetic operations on a pair of input polynomials (A_1, A_2) , whose Euclidean representation has degree pattern d . For a comprehensive account of these results see also [25].

4.2. Quantifier elimination over the reals

Semialgebraic sets were first studied in depth by Tarski [111]. One of his main results was the fact that projections of semialgebraic sets are semialgebraic. His method of proof even yielded an algorithm for computing a system of inequalities for the projection. Since projections are the geometric counterpart of existential quantifiers, their repeated use (combined with that of unions, intersections and complements) shows that any formula of the first order theory of the reals in n variables defines a semialgebraic subset of \mathbb{R}^n . Tarski's theorem can then be seen as an algorithm to "eliminate quantifiers." With the emphasis on complexity issues that developed in the 1970's, a series of quantifier elimination algorithms were proposed with increasingly better complexity [6, 33, 53, 55, 62, 101].

We next state a version of quantifier elimination over the reals taken from Renegar [101, Part III]. (The bounds in [6] are slightly better, but this will not be important for our purposes.)

In the sequel $\mathcal{F}_{\mathbb{R}}$ denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers.

Theorem 4.1. *Let F be a formula in $\mathcal{F}_{\mathbb{R}}$ in prenex form with k free variables, n bounded variables, w alternating quantifier blocks, and m atomic predicates given by real polynomials of degree at most $\delta \geq 2$. That is, F has the form*

$$\left(Q_1 x^{(1)} \in \mathbb{R}^{n_1}\right) \dots \left(Q_w x^{(w)} \in \mathbb{R}^{n_w}\right) G(y, x^{(1)}, \dots, x^{(w)})$$

with alternating quantifiers $Q_i \in \{\exists, \forall\}$ and free variables $y = (y_1, \dots, y_k) \in \mathbb{R}^k$; the quantifier free formula G is a Boolean function of m atomic predicates

$$g_j(y, x^{(1)}, \dots, x^{(w)}) \Delta_j 0, \quad 1 \leq j \leq m,$$

where the g_j are real polynomials of degree at most δ . Hereby, Δ_j is any of the standard relations $\{\geq, >, =, \neq, \leq, <\}$.

Then F is equivalent to a quantifier-free formula F' in disjunctive normal form

$$\bigvee_{i=1}^I \bigwedge_{j=1}^{J_i} (h_{ij} \Delta_{ij} 0)$$

with $M := \sum_{i=1}^I J_i$ atomic predicates involving polynomials h_{ij} of degree at most D such that

$$\log D \leq 2^{\mathcal{O}(w)} \log(m\delta) \prod_{i=1}^w n_i, \quad \log M \leq 2^{\mathcal{O}(w)} (k+1) \log(m\delta) \prod_{i=1}^w n_i.$$

The formula F' can be computed from F in parallel time (over \mathbb{R})

$$\left(2^w (k+1) \log(m\delta) \prod_{i=1}^w n_i \right)^{\mathcal{O}(1)}$$

with a total of $(m\delta)^{2^{\mathcal{O}(w)}(k+1) \prod_{i=1}^w n_i}$ real number operations (without divisions).

If, additionally, the g_j are integer polynomials with coefficients of bit size at most ℓ , then the h_{ij} can be assumed to be integer polynomials whose coefficients have bit size at most L , where $\log L \leq 2^{\mathcal{O}(w)} \log(m\delta) \prod_{i=1}^w n_i + \mathcal{O}(\log(k+\ell))$. Moreover, the formula F' can be computed from F in parallel time (classical)

$$\left(2^w (k+1) \log(m\delta) \prod_{i=1}^w n_i \right)^{\mathcal{O}(1)} \log \ell$$

with a total of $\ell^2 (m\delta)^{2^{\mathcal{O}(w)}(k+1) \prod_{i=1}^w n_i}$ bit operations.

A particular case of quantifier elimination is that for sentences, that is, formulas without free variables. In this case, after eliminating the quantifiers, one obtains a formula without variables whose truth can be easily deduced. Several questions of a basic geometric nature can be cast in this form. For instance, if $\varphi(x)$ is a system of inequalities describing a semialgebraic set W , the question “Is W open?” amounts to the truth of

$$\forall x \exists \varepsilon \forall y \left(\varepsilon > 0 \wedge (\varphi(x) \wedge \|x - y\|^2 < \varepsilon^2 \Rightarrow \varphi(y)) \right)$$

and the question “Is W bounded?” to the truth of

$$\exists B \forall x \left(B > 0 \wedge (\varphi(x) \Rightarrow \|x\|^2 < B^2) \right).$$

Theorem 4.1 implies the following upper bound on the general decision problem for formulas in $\mathcal{F}_{\mathbb{R}}$.

Corollary 4.1. *Let F be a sentence in $\mathcal{F}_{\mathbb{R}}$ as in Theorem 4.1 ($k = 0$). Then the truth of F can be decided in parallel time $\left(2^w \log(m\delta) \prod_{i=1}^w n_i\right)^{\mathcal{O}(1)}$ (over \mathbb{R}) with a total of $(m\delta)^{2^{\mathcal{O}(w)}} \prod_{i=1}^w n_i$ real number operations. In particular, an existential formula*

$$\exists x_1 \in \mathbb{R} \dots \exists x_n \in \mathbb{R} G(x_1, \dots, x_n)$$

can be decided in parallel time $(n \log(m\delta))^{\mathcal{O}(1)}$ with a total of $(m\delta)^{\mathcal{O}(n)}$ real number operations.

Let $\mathcal{F}_{\mathbb{C}}$ be the set of first order formulas over the language of the theory of fields with constant symbols for complex numbers. Results similar to those for $\mathcal{F}_{\mathbb{R}}$ hold for quantifier elimination of formulas in $\mathcal{F}_{\mathbb{C}}$ as well, see [44, 61] and the references given there.

We just state the following result about the complexity of deciding a system of polynomial equations over \mathbb{C} . It follows by a derandomization argument using witness sequences (cf. §7.1) from the randomized algebraic algorithms in [51]. In the form below it was apparently first stated in [72].

Theorem 4.2. *Let f_1, \dots, f_r be complex polynomials in n variables of degree at most $\delta \geq 2$. The truth of an existential sentence*

$$\exists x \in \mathbb{C}^n f_1(x) = 0, \dots, f_r(x) = 0$$

can be decided in parallel time $(n \log(r\delta))^{\mathcal{O}(1)}$ (over \mathbb{C}) with a total of $r^{\mathcal{O}(1)} \delta^{\mathcal{O}(n)}$ complex number operations.

4.3. Computing topological invariants of semialgebraic sets

Based on the algorithms for quantifier elimination, the following result on counting solutions and computing the dimension can be shown. Proofs can be

found for instance in [8, §13.1 & §14.4], which contain more precise bounds (however, the bounds on parallel time are not explicitly stated there). For a simple proof for the dimension problem we refer to [75].

Theorem 4.3. *Let the semialgebraic set $S \subseteq \mathbb{R}^n$ be given by a quantifier free sentence involving m real polynomials of degree at most $\delta \geq 2$.*

- (i) *One can decide whether S is finite and, if yes, compute the number of points of S in parallel time $(n \log(m\delta))^{\mathcal{O}(1)}$ with a total of $(m\delta)^{\mathcal{O}(n)}$ real number operations. If the polynomials describing S have integer coefficients of bit size at most ℓ , then this task can be performed in parallel time $(n \log(m\delta) \log \ell)^{\mathcal{O}(1)}$ with $(m\delta)^{\mathcal{O}(n)} \ell^{\mathcal{O}(1)}$ bit operations.*
- (ii) *One can compute the dimension of S in parallel time $(n \log(m\delta))^{\mathcal{O}(1)}$ with a total of $(m\delta)^{\mathcal{O}(n^2)}$ real number operations. If the polynomials describing S have integer coefficients of bit size at most ℓ , then this task can be performed in parallel time $(n \log(m\delta) \log \ell)^{\mathcal{O}(1)}$ with $(m\delta)^{\mathcal{O}(n^2)} \ell^{\mathcal{O}(1)}$ bit operations.*

Connectivity properties of semialgebraic sets are of interest in robot motion planning problems. The first algorithmic solution for these properties was given in Schwartz and Sharir [104] based on Collin’s [33] method of cylindrical algebraic decomposition. This approach allows one to compute all Betti numbers of a semialgebraic set $S \subseteq \mathbb{R}^n$. The algorithm’s complexity, however, is doubly exponential in the dimension n .

Starting with the first single exponential algorithm due to Canny [29, 30], in a series of papers [7, 31, 54, 56, 63] increasingly better algorithms for the problem to describe and to count the number of connected components of a semialgebraic set were developed. The state of the art is in Basu et al. [7] (see also [8, §16]), from which we take the following result (note, again, the bounds on parallel time are not explicitly stated there). We remark that [7] actually deals with the more general problem of computing a “roadmap” of a semialgebraic set and provides somewhat finer complexity estimates.

Theorem 4.4. *Let the semialgebraic set $S \subseteq \mathbb{R}^n$ be given by a quantifier free sentence involving m real polynomials of degree at most $\delta \geq 2$. Then*

the number of connected components of S can be computed in parallel time $(n \log(m\delta))^{\mathcal{O}(1)}$ with a total of $m^{n+1}\delta^{\mathcal{O}(n^2)}$ real number operations. If the polynomials describing S have integer coefficients of bit size at most ℓ , then this task can be performed in parallel time $(n \log(m\delta) \log \ell)^{\mathcal{O}(1)}$ with $m^{n+1}\delta^{\mathcal{O}(n^2)}\ell^{\mathcal{O}(1)}$ bit operations.

Basu [4] describes the first single exponential time algorithm for computing the Euler characteristic of a semialgebraic set S . His algorithm combines fast quantifier elimination with Morse theory and uses $(mn\delta)^{\mathcal{O}(n)}$ real number operations, or $(mn\delta)^{\mathcal{O}(n)}\ell^{\mathcal{O}(1)}$ bit operations, where S is given as above (see also [8, §13.4]).

There is not much known about the complexity to compute the higher Betti numbers of semialgebraic sets. For instance it is unknown, whether these quantities can be computed in single exponential time. For some recent results see [3].

Over the complex numbers, the following is known [50, 72].

Theorem 4.5. *Let $Z \subseteq \mathbb{C}^n$ be given as the zero set of r complex polynomials of degree at most $\delta \geq 2$. Then the dimension of Z can be computed in parallel time $(n \log(r\delta))^{\mathcal{O}(1)}$ with $r^{\mathcal{O}(1)}\delta^{\mathcal{O}(n)}$ complex number operations. Moreover, if Z is finite, the cardinality of Z can be computed within the same time bounds.*

We remark that the upper bound for computing the dimension follows from Theorem 4.2 combined with the proof of Theorem 7.3. However, for integer polynomials, the resulting algorithm is not polynomial time in the Turing model. For this situation, an algorithm for computing the dimension with $(r\ell)^{\mathcal{O}(1)}\delta^{\mathcal{O}(n)}$ bit operations was described in [32], where ℓ is an upper bound on the bit size of the coefficients of the input polynomials.

5. Concrete complexity: some lower bounds

Computational decision problems can often be cast in the following form: Given real numbers x_1, \dots, x_n , determine whether they satisfy some fixed system of polynomial equalities and inequalities. In other words, we have to decide

for a given point $x \in \mathbb{R}^n$, whether it is contained in a fixed semi-algebraic subset S_n of \mathbb{R}^n . In §4.1 we have listed some basic computational problems which can be formulated in such a way and indicated that, typically, these problems can be algorithmically solved much faster than one would naively expect. The optimality proof for these algorithms is one of the great successes of algebraic complexity theory. We present here some of the most important ideas leading to nonlinear complexity lower bounds and optimality proofs. As the underlying model of computation, we use the algebraic circuits and algebraic computation trees, introduced in §3.1 and §3.3. Note that lower bounds using these nonuniform models imply uniform lower bounds. We partly follow [24].

5.1. Geometric degree

Strassen's Degree Bound 5.1 [109] is historically the first result providing nonlinear complexity lower bounds. This fundamental insight bounds the non-scalar complexity of a set of rational functions from below by the logarithm of the degree of the graph of the associated rational map. The proof relies on Bézout's Inequality 2.1.

We consider the algorithmical problem to compute real polynomials f_1, \dots, f_m in $\mathbb{R}[X_1, \dots, X_n]$ from variables X_1, \dots, X_n (considered as the inputs) and real constants by means of straight-line programs. For simplicity of exposition, we assume that there are no divisions. A straight-line program performing this task produces a sequence $g_{-n} = 1, g_{-n+1} = X_1, \dots, g_0 = X_n, g_1, \dots, g_r$ of intermediate results such that for all $1 \leq k \leq r$ there are $i, j < k$ satisfying

$$g_k = g_i \circ g_j \text{ or } g_k = \lambda g_i, \quad \circ \in \{+, -, *\}, \lambda \in \mathbb{R},$$

and such that all f_i occur among the intermediate results. The special treatment of scalar multiplications is motivated by the lower bound we are going to exhibit, which in fact holds for the minimal number of nonscalar multiplications sufficient for such a computation. This quantity is called the (*nonscalar*) complexity $L(f_1, \dots, f_m)$ of the polynomials to be computed.

Since the degree can at most double in a multiplication step, it is obvious that $\deg g_k \leq 2^{\mu_k}$, where μ_k denotes the number of nonscalar multiplication

steps in the initial segment of the computation up to g_k . Therefore, the degree bound $L(f_m) \geq \log_2 \deg f_m$ holds. Our goal is to extend this elementary observation to the case of several polynomials.

We first note that any straight-line program solving the computation problem over the reals also works over the complex numbers, hence we may assume without loss of generality that $f_i \in \mathbb{C}[X_1, \dots, X_n]$ and that the computation takes place in the polynomial ring over \mathbb{C} .

We assign to a sequence (f_1, \dots, f_m) of polynomials the graph of the corresponding polynomial map $f: \mathbb{C}^n \rightarrow \mathbb{C}^m$ and define the degree $\deg(f_1, \dots, f_m)$ as the degree of this graph. Note that this clearly extends the usual notion of degree for polynomials ($m = 1$). What is the growth of the degrees $d_k := \deg(g_{-n+1}, \dots, g_k)$ during a straight-line computation with intermediate results g_i ? We first note that $d_0 = 1$. If $g_k = g_i * g_j$, then we can write $G_k := \text{graph}(g_{-n+1}, \dots, g_{k-1}, g_k)$ as the intersection of $G_{k-1} \times \mathbb{C}$ with the quadric given by the equation $Y_k - Y_i Y_j = 0$, where the Y_i denote the corresponding coordinate variables. Since a quadric has degree two, it follows from Bézout's Inequality 2.1 that $d_k \leq 2d_{k-1}$. In the case $g_k = \lambda g_i + \mu g_j$, $\lambda, \mu \in \mathbb{C}$, we obtain by intersecting with a linear subspace that $d_k \leq d_{k-1}$ (in fact, equality holds). We conclude that $d_k \leq 2^{\mu_k}$, where again μ_k denotes the number of nonscalar multiplication steps in the initial segment of the computation up to g_k . Finally, one can show that the degree does not increase under projections, which implies that $\deg(f_1, \dots, f_m) \leq d_r$. We therefore obtain the following fundamental result due to Strassen [109].

Degree Bound 5.1. *For polynomials f_i over \mathbb{C} we have*

$$L(f_1, \dots, f_m) \geq \log_2 \deg(f_1, \dots, f_m).$$

We remark that this lower bound remains true when allowing divisions and the computation of rational functions.

The Degree Bound 5.1 implies the optimality with respect to nonscalar complexity of numerous basic algorithms [109], see also [25]. For instance, for the elementary symmetric polynomials σ_i in n variables, it is not hard to see that $\deg(\sigma_1, \dots, \sigma_n) = n!$. The Degree Bound implies $L(\sigma_1, \dots, \sigma_n) \geq \log_2 n! \geq n(\log_2 n - 2)$, which shows the optimality (up to a constant factor)

of the corresponding algorithm mentioned in §4.1.

One of the most beautiful applications of the Degree Bound 5.1 is Strassen’s proof of the optimality of the Knuth-Schönhage algorithm (compare §4.1). To state this result let $D(d)$ denote the set of all pairs (A_1, A_2) of complex polynomials whose Euclidean representation has the degree pattern $d = (d_1, \dots, d_t)$. Strassen [110] proved that any algebraic computation tree over \mathbb{C} computing the Euclidean representation of given polynomials needs at least $n(H(d) - 2)$ nonscalar operations on all inputs (A_1, A_2) in a Zariski dense subset of $D(d)$. This almost matches the upper bound $\mathcal{O}(n(1 + H(d)))$ mentioned in §4.1.

The Degree Bound develops its full strength only in combination with the so-called Derivative Inequality due to Baur and Strassen [9], which relates the complexity of a polynomial f with the complexity of its gradient $\text{grad } f$.

Derivative Inequality 5.1. *We have $L(f, \text{grad } f) \leq 3L(f)$ for a polynomial f over \mathbb{R} or \mathbb{C} .*

In combination with the Degree Bound 5.1 we obtain the lower bound

$$(5.1) \quad L(f) \geq \frac{1}{3} \log_2 \deg(f, \text{grad } f)$$

for the complexity of a polynomial f . This implies, for instance, that the upper bound $L(\prod_{i < j} (X_i - X_j)^2) = \mathcal{O}(n \log n)$ for the discriminant as a function of the roots [9] which we saw in §4.1 is optimal up to a constant factor.

5.2. Connected components

Our goal now is to derive lower bounds on the decision complexity $C(S)$ of a semialgebraic set S in \mathbb{R}^n in terms of topological invariants of the set S . The simplest such invariant is the number $b_0(S)$ of connected components of S . If we focus on semilinear sets S and restrict the model of computation to linear decision trees, then it is an easy exercise to show that at least $\log_3 b_0(S)$ sign tests are needed for deciding membership to S (each sign test decomposes the set under consideration into at most three connected components). This lower bound was extended to the model of algebraic decision trees of higher order by Steele and Yao [107], who were the first who recognized that the Bound 5.1 below can be applied to prove nontrivial complexity lower bounds. This idea

was then taken up by Ben-Or [10] and extended to algebraic computation trees as follows.

Connected Component Bound 5.1. *The decision complexity of a semialgebraic set S in \mathbb{R}^n satisfies*

$$C(S) \geq \frac{1}{2}(\log_3 b_0(S) - n).$$

This result implies lower bounds of order $\Omega(n \log n)$ for the Element Distinctness or the Convex Hull problem and thus proves the optimality of the algorithms for these problems mentioned in §4.1. For the Knapsack problem, one obtains the lower bound $\Omega(n^2)$. For the determination of the number of connected components in these cases we refer to [42] or [25, Chap. 11].

The proof of the Connected Component Bound 5.1 relies on the fundamental bounds on the Betti numbers of real algebraic varieties due to Oleĭnik and Petrovski [95], Oleĭnik [94], Milnor [87], and Thom [112]. It will be convenient to denote the sum of all Betti numbers of a semialgebraic set S by $b(S) := \sum_{k \in \mathbb{N}} b_k(S)$. The following bound on $b(S)$ given by Milnor [87] is particularly useful for us.

Milnor-Oleĭnik-Petrovski-Thom Bound 5.1. *Assume S is the zero set of the polynomials $f_1, \dots, f_r \in \mathbb{R}[X_1, \dots, X_n]$ of degree at most $d \geq 1$. Then we have*

$$b(S) \leq d(2d - 1)^{n-1}.$$

The proof of this bound involves the Bézout Inequality 2.1 and Morse theory as explained in §7.3. We remark that the ideas of the proof are used in the fast algorithms for quantifier elimination over the reals mentioned in §4.2 (critical points method).

Corollary 5.1. *Assume that the semialgebraic set S in \mathbb{R}^n is given by the conditions $f_1 = 0, \dots, f_r = 0, g_1 \geq 0, \dots, g_s \geq 0, h_1 > 0, \dots, h_t > 0$, where f_i, g_j, h_k are polynomials of degree at most $d \geq 1$. Then we have $b(S) \leq d(2d - 1)^{n+s+t-1}$.*

PROOF – Without loss of generality we may assume $d \geq 2$. Assume first that $t = 0$. We consider the zero set $Z \subseteq \mathbb{R}^{n+s}$ of the polynomials f_1, \dots, f_r ,

$g_1 - Y_1^2, \dots, g_s - Y_s^2$, where the Y_j are additional variables. The projection $\pi: Z \rightarrow S, (x, y) \mapsto x$ has the section $S \rightarrow Z, s(x) = (x, \sqrt{g_1(x)}, \dots, \sqrt{g_s(x)})$, therefore $\pi_*: H_*(Z; \mathbb{Q}) \rightarrow H_*(S; \mathbb{Q})$ is surjective. Theorem 5.1 implies that $b(S) \leq b(Z) \leq d(2d-1)^{n+s-1}$.

If $t > 0$ then we replace the strict inequalities $h_k > 0$ by weak ones $h_k \geq \epsilon$ obtaining the semialgebraic set S_ϵ for $\epsilon > 0$. According to the previous discussion we have $b(S_\epsilon) \leq b(Z) \leq d(2d-1)^{n+s+t-1}$. The set S is the monotone union of the subspaces S_ϵ and each compact subset of S is contained in some S_ϵ . Therefore, $H_*(S; \mathbb{Q})$ is the direct limit of the $H_*(S_\epsilon; \mathbb{Q})$ (cf. [59, p. 244]). This implies the assertion. (Note that by Hardt's triviality theorem [8, Theorem 5.46] the S_ϵ are in fact homeomorphic for small enough $\epsilon > 0$.) \square

Proof of the Connected Component Bound 5.1

PROOF – We are going to bound the number of connected components of leaf sets of computation trees. Fix a leaf v and consider the corresponding path in the tree. Forgetting for the moment about the test instructions, this path defines a straight-line program with intermediate results $g_{-n} = 1, g_{-n+1} = X_1, \dots, g_0 = X_n, g_1, \dots, g_r$. Similarly as in the proof of the Degree Bound 5.1, we can describe the graph of (g_{-n+1}, \dots, g_r) by a system of linear and quadratic equations in the variables $X_1, \dots, X_n, Y_1, \dots, Y_r$. We can make this description more concise by eliminating the Y -variables belonging to linear operations. Then the graph is homeomorphic to a subset of \mathbb{R}^{n+m} given by quadratic equations, where m denotes the number of multiplication instructions along the path. By adding the linear inequalities corresponding to the sign tests, we obtain a subset of \mathbb{R}^{n+m} , which is homeomorphic to the leaf set D_v . If we assume that there are t sign tests along the path of v , then we obtain from Corollary 5.1 that $b_0(D_v) \leq 2 \cdot 3^{n+m+t-1}$.

Suppose now that an algebraic computation tree of depth D solves the membership problem to the set S in \mathbb{R}^n . Then S is the union of all the leaf sets corresponding to yes-leaves. Moreover, we have the estimate $m + t \leq D$ for each path. The number of connected components behaves subadditively with respect to the union of sets. Therefore, as there are at most 3^D leaves, we get

$$b_0(S) \leq \sum_v b_0(D_v) \leq 3^D \cdot 2 \cdot 3^{n+D-1} \leq 3^{n+2D},$$

which implies $D \geq \frac{1}{2}(\log_3 b_0(S) - n)$ and thus finishes the proof. \square

Remark 5.1 -

- (i) The proof shows that the Connected Component Bound 5.1 actually holds for the number of nonscalar multiplications and branchings. Thus additions, subtractions, and multiplications with real scalars are not counted. The same is true for all other lower bounds on $C(S)$ discussed in this section.
- (ii) For more recent bounds on the Betti numbers of semialgebraic sets we refer to [2, 5, 46].

5.3. Modified Euler characteristic

Since the Bound 5.1 is valid for the sum of all Betti numbers, it is natural to ask whether the Connected Component Bound 5.1 can be extended correspondingly. This is in fact possible, but it took quite a while before this extension was fully developed and this development took place in several steps. First, the Connected Component Bound 5.1 was extended to the modified Euler characteristic for semilinear sets S in the model of linear decision trees by Björner et al. [14]. Then Yao [118] generalized this results to the model of algebraic computation trees. We are going to describe his result in the following.

To motivate the notion of the modified Euler characteristic, note that in general $\chi(S) \neq \chi(S_1) + \chi(S_2)$ for a disjoint union S of two semialgebraic sets S_1 and S_2 . For instance, consider the closed 3-dimensional unit ball D^3 decomposed into its interior e^3 and its boundary S^2 . Then $\chi(D^3) = \chi(e^3) = 1$ but $\chi(S^2) = 2$.

Yao [118] defined the *modified Euler characteristic* χ^* of semialgebraic sets, which satisfies an additivity property, and coincides with the usual Euler characteristic for compact semialgebraic sets. The following proposition from [118] characterizes this quantity. We remark that its proof of existence relies on Hironaka's triangulation theorem [64] for bounded (not necessarily closed) semialgebraic sets.

Proposition 5.1. *There is a unique function χ^* mapping semialgebraic sets to integers, which satisfies the following properties:*

- (i) If $S = \bigsqcup_{i=1}^N S_i$ is a disjoint union of semialgebraic sets then $\chi^*(S) = \sum_{i=1}^N \chi^*(S_i)$.
- (ii) If S is a compact semialgebraic set then $\chi^*(S) = \chi(S)$.
- (iii) If there is a semialgebraic homeomorphism $S \xrightarrow{\sim} T$ then $\chi^*(S) = \chi^*(T)$.

Example 5.1 - The stereographic projection

$$S^n - \{(0, \dots, 0, 1)\} \xrightarrow{\sim} \mathbb{R}^n, x \mapsto y$$

given by the equations $y_i = x_i/(1 - x_{n+1})$, is a homeomorphism. Hence $\chi^*(e^n) = \chi^*(\mathbb{R}^n) = \chi(S^n) - 1 = (-1)^n$. Note that, in contrast with χ , χ^* is not invariant under homotopies.

In the next section we will derive the following result due to Yao [118] from a more general lower bound in terms of Borel-Moore Betti numbers.

Euler Characteristic Bound 5.1. *The decision complexity of a semialgebraic set S in \mathbb{R}^n satisfies $C(S) \geq \frac{1}{3}(\log_3 \chi^*(S) - n - 4)$.*

5.4. Borel-Moore Betti numbers

The Euler Characteristic Bound 5.1 was generalized to Borel-Moore Betti numbers of semilinear sets S in the model of linear decision trees by Björner and Lovász [13], and finally generalized by Yao [119] to the general case of semialgebraic sets. We are going to discuss this next. The difficulty is that the higher Betti numbers do not behave subadditively with respect to disjoint unions (see Figure 1). The key idea is to replace the Betti numbers by a related

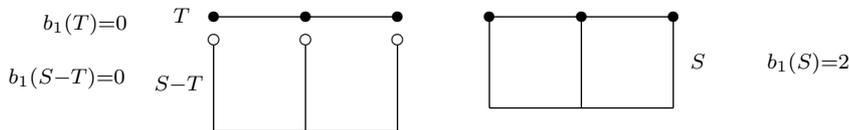


Figure 1: Betti numbers are not subadditive.

quantity which behaves subadditively. This is achieved by working with the

Borel-Moore homology [19]. We say that a subset S of \mathbb{R}^n is *locally closed* if it is the intersection of an open with a closed subset of \mathbb{R}^n .

Let S be a locally closed semialgebraic subset of \mathbb{R}^n . If S is not compact, we may compactify it by “adding a point at infinity”. Formally, the *Alexandrov compactification* of S is a pair (\dot{S}, ι) such that \dot{S} is a compact semialgebraic set, $\iota: S \rightarrow \dot{S}$ is a continuous semialgebraic map, which is a homeomorphism onto its image, and $\dot{S} - \iota(S)$ consists of just one point, denoted by ∞ . One can show that this object exists and that it is essentially unique (cf. [18]). If S is closed, then one may take for ι the restriction to S of the inverse of the stereographic projection $S^n - \{(0, \dots, 0, 1)\} \xrightarrow{\sim} \mathbb{R}^n$.

Let F be a field and S be as above. If S is not compact, then the *Borel-Moore homology vector spaces* of S over F are defined as the relative homology spaces of the pair (\dot{S}, ∞) , that is, $H_k^{\text{BM}}(S; F) := H_k(\dot{S}, \infty; F)$, cf. [18, §11.4]. If S is compact, then we define $H_k^{\text{BM}}(S; F) = H_k(S; F)$. Moreover, we define the *Borel-Moore Betti numbers* $b_k^{\text{BM}}(S)$ of S by

$$b_k^{\text{BM}}(S) := \dim H_k^{\text{BM}}(S; \mathbb{Q}), \quad b^{\text{BM}}(S) := \sum_{k \in \mathbb{N}} b_k^{\text{BM}}(S).$$

Note that for noncompact S , we have $b_k^{\text{BM}}(S) := b_k(\dot{S})$ for $k > 0$ and $b_0^{\text{BM}}(S) = b_0(\dot{S}) - 1$. Clearly, $b_k^{\text{BM}}(S) = b_k(S)$ for compact S and $k \geq 0$.

Proposition 5.2. *We have $\chi^*(S) = \sum_{k \geq 0} (-1)^k b_k^{\text{BM}}(S)$ for a locally closed semialgebraic set S .*

PROOF – If S is compact the result is trivial. Otherwise, by additivity of χ^* , we have $\chi^*(S) = \chi^*(\dot{S}) - \chi^*(\infty) = \chi(\dot{S}) - 1 = \chi(\dot{S}, \infty)$. On the other hand

$$\chi(\dot{S}, \infty) = \sum_k (-1)^k \dim H_k(\dot{S}, \infty; \mathbb{Q}) = \sum_k (-1)^k b_k^{\text{BM}}(S),$$

which shows the assertion. \square

For our purposes, the subadditivity property of the Borel-Moore Betti numbers stated in the next lemma is crucial.

Lemma 5.1. *Let S, T be locally closed, semialgebraic sets such that T is a closed subset of S . Then $b_k^{\text{BM}}(S) \leq b_k^{\text{BM}}(S - T) + b_k^{\text{BM}}(T)$ for all $k \in \mathbb{N}$.*

The reader might illustrate this for the example in Figure 1: $b(T) = b^{\text{BM}}(T) = 1$, $b(S) = b^{\text{BM}}(S) = 3$, $b(S - T) = 1$, however $b^{\text{BM}}(S - T) = b(S) - 1 = 2$, as S is homeomorphic to the Alexandrov compactification of $S - T$.

Proof of Lemma 5.1

PROOF – We will notationally omit the reference to the base field \mathbb{Q} for ease of notation. We need the following characterization of the Borel-Moore homology: if $T \subseteq S$ are compact, semialgebraic subsets, then we have

$$(5.2) \quad H_k^{\text{BM}}(S - T) \simeq H_k(S, T).$$

In order to show this, we may assume that $U := S - T$ is not closed. Let $\iota: U \rightarrow \dot{U}$ be the Alexandrov compactification of U . We extend this map to S by setting $\iota(t) = \infty$ for all $t \in T$. Then it is not hard to see that the extended map $S \rightarrow \dot{U}$ is in fact the quotient S/T of the topological space S obtained by collapsing T to a point $[T]$. Hence we obtain $H_k^{\text{BM}}(U) \simeq H_k(\dot{U}, \infty) \simeq H_k(S/T, [T])$. Since (pairs of) semialgebraic sets possess cellular decompositions (cf. [18]), we conclude that $H_k(S/T, [T]) \simeq H_k(S, T)$ by using a standard fact of algebraic topology (cf. [102, Thm. 8.41]). This proves the claim (5.2).

Let $S \supseteq T \supseteq R$ be a triple of topological spaces. The corresponding long exact sequence of homology

$$\cdots \rightarrow H_k(T, R) \rightarrow H_k(S, R) \rightarrow H_k(S, T) \rightarrow H_{k-1}(T, R) \rightarrow \cdots$$

implies that

$$(5.3) \quad \dim H_k(S, R) \leq \dim H_k(S, T) + \dim H_k(T, R).$$

In order to show the assertion of Lemma 5.1, assume first that both S and T are not compact. Applying (5.3) to the triple $\dot{S} \supseteq \dot{T} \supseteq \{\infty\}$ and using (5.2) we get

$$\begin{aligned} b_k^{\text{BM}}(S) = \dim H_k(\dot{S}, \infty) &\leq \dim H_k(\dot{S}, \dot{T}) + \dim H_k(\dot{T}, \infty) \\ &= \dim H_k^{\text{BM}}(\dot{S} - \dot{T}) + b_k^{\text{BM}}(T) \\ &= b_k^{\text{BM}}(S - T) + b_k^{\text{BM}}(T). \end{aligned}$$

The other cases can be settled similarly. \square

The Connected Component Bound 5.1 was generalized by Yao [119] as follows.

Betti Number Bound 5.1. *The decision complexity of a locally closed, semi-algebraic set S in \mathbb{R}^n satisfies*

$$C(S) \geq \frac{1}{3} (\log_3 b^{\text{BM}}(S) - n - 4) .$$

For the proof, we need an extension of Corollary 5.1 to Borel-Moore Betti numbers.

Corollary 5.2. *Consider the locally closed, semialgebraic set S in \mathbb{R}^n given by*

$$(5.4) \quad f_1 = 0, \dots, f_r = 0, g_1 \geq 0, \dots, g_s \geq 0, h_1 > 0, \dots, h_t > 0,$$

where f_i, g_j, h_k are polynomials of degree at most $d \geq 1$ and we assume that the degree of h_k is strictly less than d . Then $|\chi^*(S)| \leq b^{\text{BM}}(S) \leq d(2d-1)^{n+s+2t+1}$.

PROOF – We follow [93]. Without loss of generality we may assume that $d \geq 2$.

We first treat the case where $\mathcal{Z}(f_1, \dots, f_r)$ is bounded and $t \geq 1$. By relaxing the inequalities $h_k > 0$ to $h_k \geq 0$ in (5.4) we obtain a compact set A and we can write $S = A - B$ with the compact set $B := A \cap \mathcal{Z}(h_1 \cdots h_t)$. The long exact sequence of homology $\cdots \rightarrow H_k(B) \rightarrow H_k(A) \rightarrow H_k(A, B) \rightarrow H_{k-1}(B) \rightarrow \cdots$ implies that

$$\dim H_k(A, B) \leq \dim H_k(A) + \dim H_{k-1}(B).$$

On the other hand, by (5.2), we have $H_k^{\text{BM}}(A - B) \simeq H_k(A, B)$. This implies

$$b^{\text{BM}}(S) = \sum_k \dim H_k(A, B) \leq b(A) + b(B).$$

Corollary 5.1 implies that $b(A) \leq d(2d-1)^{n+s+t-1}$. In order to estimate $b(B)$ we consider the set (put $y_0 := 1$)

$$B' := \{(x, y) \in \mathbb{R}^{n+t-1} \mid x \in A, y_i = y_{i-1}h_i(x) \text{ for } 1 \leq i < t, y_{t-1}h_t(x) = 0\},$$

which is homeomorphic to B . Corollary 5.1 applied to B' yields $b(B) = b(B') \leq d(2d-1)^{n+t-1+s+t-1}$ (recall that we assume $\deg h_k < d$). Altogether, we obtain $b^{\text{BM}}(S) \leq 2d(2d-1)^{n+s+2t-2}$.

If $\mathcal{Z}(f_1, \dots, f_r)$ is unbounded, then we replace S by its inverse image S' under the stereographic projection $S^n - \{(0, \dots, 0, 1)\} \xrightarrow{\sim} \mathbb{R}^n$. The set $S' \subseteq \mathbb{R}^{n+1}$ is bounded and given by the two additional constraints $\sum_{k=1}^{n+1} x_k^2 = 1$, $x_{n+1} < 1$ besides the (in)equalities arising from (5.4) by transformation. (For instance, $f_i(x) = 0$ transforms to $(1 - x_{n+1})^d f_i(x/(1 - x_{n+1})) = 0$.) The claim follows by applying the result of the previous case to S' (with n and t increased by 1). \square

Proof of the Betti Number Bound 5.1

PROOF – The proof is completely analogous to the proof of the Connected Component Bound 5.1. The only remaining issue is to prove that indeed $b_k^{\text{BM}}(S) \leq \sum_v b_k^{\text{BM}}(D_v)$ for an algebraic computation tree deciding membership to a locally closed, semialgebraic set S . To a node u of such a tree we associate the semialgebraic set S_u consisting of the points $x \in S$ whose path in the tree passes through u . It is obvious that S_u is locally closed. Let L_u denote the set of **yes**-leaves corresponding to paths passing through u . Then we have $S_u = \cup_{v \in L_u} D_v$. We prove now by reverse induction on the depth of u that

$$(5.5) \quad b_k^{\text{BM}}(S_u) \leq \sum_{v \in L_u} b_k^{\text{BM}}(D_v).$$

If u is a leaf, then there is nothing to show. Otherwise, let u be a node with the descendents u_-, u_0, u_+ corresponding to the outcome of the sign test with a polynomial f , thus $S_{u_-} = S \cap \{f < 0\}$, $S_{u_0} = S \cap \{f = 0\}$, $S_{u_+} = S \cap \{f > 0\}$. Since S_{u_0} is closed in $S_{u_-} \cup S_{u_0}$ and this set is closed in S_u , we may apply Lemma 5.1 twice in order to obtain $b_k^{\text{BM}}(S_u) \leq b_k^{\text{BM}}(S_{u_-}) + b_k^{\text{BM}}(S_{u_0}) + b_k^{\text{BM}}(S_{u_+})$. This shows the claim (5.5) and completes the proof of the Betti Number Bound 5.1. \square

Remark 5.2 - The Euler Characteristic Bound 5.1 follows for locally closed sets S from the statement of the Betti Number Bound 5.1. However, since

the modified Euler characteristic is additive for any semialgebraic sets (Proposition 5.1), the above proof shows that the Euler Characteristic Bound 5.1 actually holds for arbitrary semialgebraic sets.

A nice application of the Betti Number Bound 5.1 is the optimal lower bound $\Omega(n \log \frac{n}{k})$ for the k -equal problem, mentioned in §4.1, cf. Björner et al. [13, 14]. For the k -equal problem, the computation of Betti numbers is done using a rich theory, focusing on the homology of subspace arrangements and the formula of Goresky and MacPherson [52], which characterizes the Betti numbers of the complement of a subspace arrangement by its intersection semi-lattice. We refer to the excellent survey by Björner [12] for more information on this.

We remark that the computation of (Borel-Moore) Betti numbers for concrete examples is a highly nontrivial task (cf. §4.3). A complexity theorist's explanation of this empirical fact is given in §7 of this survey.

Remark 5.3 - Here is another application of the Betti Number Bound 5.1. Let $Z \subseteq \mathbb{P}^m(\mathbb{C})$ be the complex projective zero set of the homogeneous polynomial f of degree d in $m + 1$ variables. If Z is smooth, then it is known that (cf. [41])

$$\chi(Z) = m + 1 + \frac{1}{d} \left((1-d)^{m+1} - 1 \right), \quad b(Z) = m + \left(1 - \frac{1}{d} \right) \left((d-1)^m + (-1)^{m+1} \right).$$

Hence both $\log \chi(Z)$ and $\log b(Z)$ have order of magnitude $\Omega(m \log d)$. For any compact semialgebraic set S embedded in some \mathbb{R}^n , which is homomorphic to Z , we obtain by both the Euler Characteristic Bound 5.1 and the Betti Number Bound 5.1 that $C(S) = \Omega(m \log d)$. For instance this is optimal, up to a constant factor, for the power sum $f = X_0^d + \dots + X_m^d$, which has a smooth projective zero set. Clearly, this lower bound cannot hold for the polynomial $g = X_0 X_1 \dots X_m$, which has complexity $\mathcal{O}(m)$. The reason why this fails is that the zero set of g is not smooth (in fact, it contains lots of singularities). This observation suggests that there might be a connection between complexity and singularities. The fact that the above lower bound avoids the use of the Derivative Inequality 5.1 also points in this direction. It is an interesting question to evaluate how the Betti Number Bound 5.1 performs on problems traditionally treated by Degree Bounds.

Remark 5.4 -

- (i) The Borel-Moore Betti numbers of S may be alternatively defined as the dimension of the cohomology groups $H_c^*(S)$ of S with compact supports, a notion naturally occurring in the Poincaré duality theorem for noncompact manifolds, cf. [59, §3.3, p. 242].
- (ii) For a complex algebraic variety Z we have $\chi^*(Z) = \chi(Z)$. If Z is smooth of complex dimension n , then this follows from the Poincaré duality $H_k(Z) \simeq H_c^{2n-k}(Z)$, using the interpretation of $\chi^*(Z)$ as the Euler characteristic of the cohomology $H_c^*(Z)$ with compact support. For the proof of the general case see [45, Exercise §4.5, p. 95 and Notes §4.13, p. 141].

5.5. Parallel complexity

The Betti Number Bound 5.1 can be extended to parallel complexity. For the number of connected components, this was done already very early by Yao [117], who proved a time-space tradeoff for the knapsack problem. The following bound was obtained by Montaña and Pardo [89].

Parallel Betti Number Bound 5.1. *The parallel decision complexity of a locally closed, semialgebraic set S in \mathbb{R}^n satisfies*

$$D(S) \geq \Omega\left(\sqrt{\frac{\log b^{\text{BM}}(S)}{n}}\right).$$

We remark that a similar lower bound in terms of the geometric degree of algebraic sets over \mathbb{C}^n was proved in [90].

For some applications, the following elementary degree bound from [22] on the parallel complexity to decide membership to hypersurfaces suffices. The lemma can be found in slightly varying form in several places in the literature [28, 35, 82, 90].

Lemma 5.2. *Let Z be an irreducible hypersurface in \mathbb{R}^n with irreducible generator g of degree d . Then any algebraic circuit \mathcal{C} deciding membership of points in \mathbb{R}^n to Z has depth at least $\log_2 d$.*

PROOF – Let \mathcal{C} be a decisional algebraic circuit solving the membership problem to Z . For simplicity, we assume \mathcal{C} to be division-free. Let β be a map which assigns to the sign gates of \mathcal{C} a value in $\{0, 1\}$. We denote by D_β the set of all inputs $x \in \mathbb{R}^n$ such that upon execution of \mathcal{C} on x , the sign gates of \mathcal{C} evaluate to the values prescribed by β . Note that either for all $x \in D_\beta$ the circuit \mathcal{C} accepts x , or for all $x \in D_\beta$ it rejects x . Thus the set Z is a union of certain D_β . As Z is irreducible there is some $D = D_\beta$ which is Zariski-dense in Z . This set D is described by conditions

$$f_1 \geq 0, \dots, f_r \geq 0, f_{r+1} < 0, \dots, f_s < 0.$$

Each polynomial f_i is computed by the algebraic circuit $\mathcal{C}_{\beta,i}$ obtained from \mathcal{C} by replacing the sign nodes by constant nodes with values according to β . We may assume without loss of generality that the f_i are not the zero polynomials. It follows that $\text{depth}(\mathcal{C}) \geq \text{depth}(\mathcal{C}_{\beta,i}) \geq \log_2 \deg f_i$ for all i , using an (easy to prove) observation in [77].

Since $\dim D < n$, we have $D \subseteq \bigcup_i \mathcal{Z}(f_i)$. Since D is Zariski dense in Z and Z is irreducible, there must be some i such that $Z \subseteq \mathcal{Z}(f_i)$. Because the vanishing ideal of Z is generated by the irreducible generator g of Z [18, p. 85], f_i must be a multiple of g and therefore $\deg f_i \geq \deg g = d$. Altogether $\text{depth}(\mathcal{C}) \geq \log_2 d$. \square

As an interesting application of Lemma 5.2, we show now an exponential lower bound on the parallel complexity of the decisional version of the quantifier elimination problem, thus complementing the exponential upper bound for the parallel time in Corollary 4.1.

We shall construct a sequence of formulas Φ_0, Φ_1, \dots in two free (real) variables z and t , the meaning of Φ_n being “ t is a 2^{2^n} -th root of z ”. The basic idea is to simulate repeated squaring by small size formulas, as in [40, 43, 60]. Put $e_n := 2^{2^n}$.

The polynomial $z - t^{e_n}$ has dense size $\mathcal{O}(e_n)$ and sparse size $\mathcal{O}(2^n)$. The goal is to describe its zero set with a formula $\Phi_n(t, z)$ of size linear in n . To do so, note that

$$z = t^{e_n} \iff \exists y (z = y^{e_{n-1}} \wedge y = t^{e_{n-1}}).$$

One could then take for $\Phi_0(t)$ the formula $z = t^2$ and recursively define

$$\Phi_n(t, z) := \exists y (\Phi_{n-1}(t, y) \wedge \Phi_{n-1}(y, z)).$$

But, when expanded, this formula has exponential size. We may avoid this explosion by noting that the two terms in the conjunction are occurrences of the same formula (just with different variables) and take

$$\Phi_n(t, z) := \exists y \forall v \forall w [(v = t \wedge w = y) \vee (v = y \wedge w = z) \Rightarrow \Phi_{n-1}(v, w)].$$

When expanded, $\Phi_n(t, z)$ is a formula whose length is linear in n and logically equivalent to $z = t^{e_n}$.

Applying Lemma 5.2 to the zero set of $g = z - t^{e_n}$, we get the lower bound $\log e_n = 2^n$ on the parallel time necessary to decide whether $\Phi_n(t, z)$ holds. Therefore, the exponential upper bound for the parallel time to decide quantified sentences in Theorem 4.1 is optimal (in the sense that the problem can not be solved in parallel polynomial time).

For another application of Lemma 5.2 to prove that the polynomial ideal membership problem over \mathbb{R} has single exponential parallel complexity, we refer to [22].

6. Structural complexity: basic classes and results

In what follows we define complexity classes and reductions over \mathbb{K} , for $\mathbb{K} = \mathbb{R}, \mathbb{C}$ or $\{0, 1\}$. To distinguish between complexity classes in these different settings we use subindices ‘ \mathbb{R} ’ and ‘ \mathbb{C} ’ for the first two settings, respectively, and none for the last (following the usual notation). In addition, to emphasize the difference between the two versions of continuous settings and the classical setting, we use **sans serif** fonts for the latter. Thus, the class of problems decidable in polynomial time in the three settings above is denoted by $P_{\mathbb{R}}$, $P_{\mathbb{C}}$ and P .

6.1. Basic decisional complexity classes

Definition 6.1. *A machine M over \mathbb{K} is said to work in polynomial time when there is a constant $c \in \mathbb{N}$ such that for every input $x \in \mathbb{K}^\infty$, M reaches*

its output node after at most $\text{size}(x)^c$ steps. The class $P_{\mathbb{K}}$ is then defined as the set of all subsets of \mathbb{K}^∞ that can be accepted by a machine working in polynomial time, and the class $FP_{\mathbb{K}}$ as the set of functions which can be computed in polynomial time. Replacing the bound $\text{size}(x)^c$ by $2^{\text{size}(x)^c}$ above one defines the classes $EXP_{\mathbb{K}}$ and $FEXP_{\mathbb{K}}$.

The classes $P_{\mathbb{R}}$ and $P_{\mathbb{C}}$ are defined above in terms of BSS machines over \mathbb{R} and \mathbb{C} , respectively. These classes can be alternatively defined in terms of algebraic circuits and classical Turing machines as follows.

Let now \mathbb{K} be \mathbb{R} or \mathbb{C} and $S \subseteq \mathbb{K}^\infty$. Recall that a family $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ of circuits decides S when the function computed by the n th circuit of the family is the restriction to \mathbb{K}^n of the characteristic function of S . We say that $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ is *P-uniform* when there exist constants $\alpha_1, \dots, \alpha_m \in \mathbb{K}$ and a deterministic Turing machine M satisfying the following: for every $n \in \mathbb{N}$, the constant gates of \mathcal{C}_n have associated constants in the set $\{\alpha_1, \dots, \alpha_m\}$ and M computes a description of the i th gate of the n th circuit in time polynomial in n (if the i th gate is a constant gate with associated constant α_k then M returns k instead of α_k).

Proposition 6.1 ([76, 97]). *Let $S \subseteq \mathbb{K}^\infty$. Then $S \in P_{\mathbb{K}}$ if and only if S can be decided by a P-uniform family of circuits with polynomial size.*

Definition 6.2. *A set A belongs to $NP_{\mathbb{K}}$ if there is a machine M satisfying the following condition: for all $x \in \mathbb{K}^\infty$, $x \in A$ iff there exists $y \in \mathbb{R}^\infty$ such that M accepts the input (x, y) within time polynomial in $\text{size}(x)$. In this case, the element y is called a witness for x .*

The set A belongs to $DNP_{\mathbb{K}}$ when we additionally require that the element y above belongs to $\{0, 1\}^\infty$.

A set A belongs to $\Sigma_{\mathbb{K}}^k$ if there is a machine M satisfying the following condition: for all $x \in \mathbb{K}^\infty$,

$$x \in A \iff \exists y_1 \in \mathbb{R}^\infty \forall y_2 \in \mathbb{R}^\infty \dots Q_k y_k \in \mathbb{R}^\infty \text{ such that } M \text{ accepts} \\ \text{the input } (x, y_1, \dots, y_k) \text{ within time polynomial in } \text{size}(x).$$

Here $Q_k = \exists$ if k is odd and \forall otherwise.

A set A is in $\Pi_{\mathbb{K}}^k$ if its complement is in $\Sigma_{\mathbb{K}}^k$. The polynomial hierarchy $\text{PH}_{\mathbb{K}}$ is the union of the classes $\Sigma_{\mathbb{K}}^k$ for $k \geq 1$. The digital polynomial hierarchy $\text{DPH}_{\mathbb{K}}$ is defined analogously.

Remark 6.1 - The element y in the definition of $\text{NP}_{\mathbb{K}}$ above can be seen as the sequence of guesses used in the Turing machine model. However, we note that in the above definition no nondeterministic machine is introduced as a computational model, and nondeterminism appears here as a new acceptance definition for the deterministic machine. Also, we note that the length of y can be easily bounded by the time bound $p(\text{size}(x))$.

It is immediate to realize that, for $\mathbb{K} = \{0, 1\}$, there is no difference between $\text{NP}_{\mathbb{K}}$ and $\text{DNP}_{\mathbb{K}}$ (and similarly for higher classes in $\text{PH}_{\mathbb{K}}$ and $\text{DPH}_{\mathbb{K}}$).

Example of problems in these classes for $\mathbb{K} = \{0, 1\}$ abound in the literature, cf. [96]. We next focus on $\mathbb{K} = \mathbb{R}$.

An example of a problem in $\text{DNP}_{\mathbb{R}}$ is linear programming feasibility. This is the problem of deciding, given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, whether there exists a point $x \in \mathbb{R}^n$ such that $Ax \leq b$. At a first glance, this problem is in $\text{NP}_{\mathbb{R}}$. But one notices that if the system $Ax \leq b$ has solutions, then there exists a subset $B \subseteq \{1, \dots, m\}$ of cardinality $\min\{n, m\}$ such that the linear system $A_B x = b_B$ has solutions and that every solution satisfies $Ax \leq b$; pick any of them (here A_B is the submatrix of A resulting from removing the rows of A whose index is not in B). The algorithm in $\text{DNP}_{\mathbb{R}}$ now follows since linear systems can be solved in polynomial time.

Examples of problems in $\text{PH}_{\mathbb{R}}$ are deciding openness (in $\Pi_{\mathbb{R}}^3$) and deciding boundedness (in $\Sigma_{\mathbb{R}}^2$) of semialgebraic sets defined in §4.2.

We can also define complexity classes measuring cost by parallel time. We denote by $\text{PAR}_{\mathbb{R}}$ the class of decision problems whose characteristic function can be computed in parallel polynomial time, i.e., by a $\text{P}_{\mathbb{R}}$ -uniform family of circuits such that $\text{depth}(\mathcal{C}_n) = n^{\mathcal{O}(1)}$. Also, $\text{FPAR}_{\mathbb{R}}$ denotes the class of functions f which can be computed with such resources, and for which there is a polynomial p such that $\text{size}(f(x)) = p(\text{size}(x))$ for all $x \in \mathbb{R}^{\infty}$.

Remark 6.2 -

- (i) Over $\{0, 1\}$, the class of sets decidable in parallel polynomial time coincides with that of the sets decidable in polynomial space [20]. For this reason, this class is usually denoted by PSPACE (and the corresponding class of functions by FPSPACE).
- (ii) As in Proposition 6.1, $P_{\mathbb{R}}$ -uniformity can be replaced by P-uniformity in the definition of $PAR_{\mathbb{R}}$.

Theorem 6.1. *The arrows in the following diagram are inclusions of complexity classes:*



PROOF – All the inclusions are easy to show except for $PH_{\mathbb{R}} \subseteq PAR_{\mathbb{R}}$, which follows from Corollary 4.1. \square

The following separation result is from [35].

Theorem 6.2. *We have $PAR_{\mathbb{R}} \neq EXP_{\mathbb{R}}$.*

PROOF – Consider the set $S = \sqcup_{n \geq 0} S_n$ where $S_n = \{x \in \mathbb{R}^n \mid x_2 = x_1^{2^{2^n}}\}$. It is clear that $S \in EXP_{\mathbb{R}}$. Lemma 5.2 implies that $D(S_n) \geq 2^n$, hence $S \notin PAR_{\mathbb{R}}$. \square

We remark that $PAR_{\mathbb{C}} \neq EXP_{\mathbb{C}}$ by a similar argument. However, the truth of the corresponding statement over $\{0, 1\}$ is unknown.

6.2. Some known completeness results for decisional problems

We first recall the basic notions of reduction for classes of decision problems. These definitions are for a class \mathcal{C} such that $P_{\mathbb{K}} \subseteq \mathcal{C}$.

Definition 6.3.

1. Let $S, T \subseteq \mathbb{K}^\infty$. We say that $\varphi: \mathbb{K}^\infty \rightarrow \mathbb{K}^\infty$ is a reduction from S to T if φ can be computed in polynomial time and, for all $x \in \mathbb{K}^\infty$, $x \in S$ if and only if $\varphi(x) \in T$.
2. We say that S Turing reduces to T if there exists an oracle machine which, with oracle T , decides S in polynomial time.
3. Let \mathcal{C} be any class of subsets of \mathbb{R}^∞ . We say that a set T is hard for \mathcal{C} if, for every $S \in \mathcal{C}$, there is a reduction from S to T . We say that T is \mathcal{C} -complete if, in addition, $T \in \mathcal{C}$.
4. The notions of Turing-hardness or Turing-completeness are defined similarly.

If a \mathcal{C} -complete problem belongs to $P_{\mathbb{K}}$ then $\mathcal{C} = P_{\mathbb{K}}$. Since it is known that $P_{\mathbb{K}} \neq \text{EXP}_{\mathbb{K}}$, a proof of $\text{EXP}_{\mathbb{K}}$ -completeness is a certificate of intractability. But $\text{EXP}_{\mathbb{K}}$ -completeness is a rare phenomenon. Instead, in the early 1970's, Levin [81] and Cook [34] independently proved that the problem SAT of deciding whether a propositional formula has a satisfying assignment is NP-complete. This result was followed by a paper by Karp [67] proving NP-completeness for 21 problems from diverse areas of mathematics and, subsequently, by an avalanche of NP-completeness results, see [47]. The size and variety of the collection of known NP-complete problems is considered today as evidence of the intractability of NP.

In this respect, the situation over \mathbb{R} or \mathbb{C} is different. One may say that, if NP-completeness exhibits a single problem with different dresses, the wardrobe of that problem in the real or complex settings seems to be definitely smaller than that in the discrete setting.

We will consider algebraic or semialgebraic sets as input data for machines over \mathbb{R} or \mathbb{C} , or as inputs for Turing machines, depending on the model we are working in. To fix ideas we will, unless otherwise specified, assume that semialgebraic sets are given as unions of basic semialgebraic sets as in (2.1). These sets are described by a family of polynomials. So, properly speaking, the input data is not the set itself but the description of it.

We also have to define how polynomials themselves are encoded. To fix ideas, we will assume that polynomials are given by the *sparse encoding* as follows. A polynomial $f = \sum_{e \in I} u_e x_1^{e_1} \cdots x_n^{e_n}$ is represented in the sparse encoding by a list of the pairs (u_e, e) for $e \in I$, where $I = \{e \in \mathbb{N}^n \mid u_e \neq 0\}$. The exponent vector e is thought to be given by a bit vector of length at most $\mathcal{O}(n \log \deg f)$. Depending on the model we are working in, the coefficients u_e are either given as real numbers, complex numbers, or by a bit vector when $u_e \in \mathbb{Z}$ and we are working in the Turing model. The sparse size of f is defined to be $|I|$ times the maximal size of (u_e, e) and the sparse size of a set of polynomials is defined as the sum of the sparse sizes of its elements.

Another way of encoding polynomials is the *dense encoding*. Here, a polynomial of degree d in n variables is given by the list of its $\binom{n+d}{d}$ coefficients. Yet another way is to encode the polynomial by a straight-line program computing it. In this case, the size of the encoding of f is the length of the straight-line program.

The following problems describing variants of the basic feasibility problem over \mathbb{R} and \mathbb{C} were introduced and studied in [17].

$\text{HN}_{\mathbb{C}}$ (*Hilbert's Nullstellensatz*) Given a finite set of multivariate complex polynomials, decide whether these polynomials have a common complex zero.

$\text{FEAS}_{\mathbb{R}}$ (*Polynomial feasibility*) Given a multivariate real polynomial, decide whether it has a real root.

$\text{SAS}_{\mathbb{R}}$ (*Semialgebraic satisfiability*) Given a semialgebraic set S , decide whether it is nonempty.

Remark 6.3 - To fix ideas, we assume in the definition of the above problems that the input polynomials are given in sparse representation. However, note that choosing the dense encoding leads to polynomial time equivalent problems. In order to see this, one just has to introduce additional variables that help to represent monomials of high degree by “repeated squaring”. The solution set of the new system of polynomial (in)equalities is homeomorphic to the original one. A similar remark applies for the encoding of polynomials by division free straight-line programs.

In [17] the following fundamental completeness result was proved.

Theorem 6.3. *The problem $\text{HN}_{\mathbb{C}}$ is $\text{NP}_{\mathbb{C}}$ -complete, and the problems $\text{FEAS}_{\mathbb{R}}$ and $\text{SAS}_{\mathbb{R}}$ are $\text{NP}_{\mathbb{R}}$ -complete.*

Consider the following decision problems related to the computation of the dimension of algebraic and semialgebraic sets.

$\text{DIM}_{\mathbb{C}}$ (*Algebraic dimension*) Given a finite set of multivariate complex polynomials with affine zero set Z and $d \in \mathbb{N}$, decide whether $\dim Z \geq d$.

$\text{DIM}_{\mathbb{R}}$ (*Semialgebraic dimension*) Given a semialgebraic set S and $d \in \mathbb{N}$, decide whether $\dim S \geq d$.

Koiran [72, 75] significantly extended the list of known geometric $\text{NP}_{\mathbb{C}}$ - or $\text{NP}_{\mathbb{R}}$ -complete problems by showing the following.

Theorem 6.4. *The problems $\text{DIM}_{\mathbb{C}}$ and $\text{DIM}_{\mathbb{R}}$ are complete in $\text{NP}_{\mathbb{C}}$ and $\text{NP}_{\mathbb{R}}$ respectively.*

We will give the completeness proof for $\text{DIM}_{\mathbb{C}}$ later (see Theorem 7.3).

6.3. Counting complexity classes

Problems in $\text{NP}_{\mathbb{K}}$ require deciding the existence of moderately small solutions for a variety of situations. A different kind of problems arise when, instead of deciding existence, one is required to count how many such solutions are there.

Definition 6.4. *We say that a function $f: \mathbb{K}^{\infty} \rightarrow \mathbb{N} \cup \{\infty\}$ belongs to the class $\#\text{P}_{\mathbb{K}}$ if there exists a polynomial time machine M over \mathbb{K} and a polynomial p such that, for all $x \in \mathbb{K}^n$,*

$$f(x) = |\{y \in \mathbb{K}^{p(n)} \mid M \text{ accepts } (x, y)\}|.$$

The complexity class $\text{FP}_{\mathbb{K}}^{\#\text{P}_{\mathbb{K}}}$ consists of all functions $f: \mathbb{K}^{\infty} \rightarrow \mathbb{K}^{\infty}$, which can be computed in polynomial time using oracle calls to functions in $\#\text{P}_{\mathbb{K}}$.

The counting class $\#P := \#P_{\{0,1\}}$ was introduced by Valiant [114, 115]. The version $\#P_{\mathbb{R}}$ over the reals was first considered by Meer in [83].

The following important result by Toda [113] shows that $\#P$ has at least the power of the polynomial hierarchy.

Theorem 6.5. *We have $PH \subseteq P^{\#P}$.*

It is an interesting open problem whether versions of Toda's result hold over \mathbb{R} or \mathbb{C} .

We next locate the newly defined counting complexity classes over $\mathbb{K} = \mathbb{R}$ within the landscape of known complexity classes.

Theorem 6.6.

- (i) *If $f \in \#P_{\mathbb{R}}$ then, for all $x \in \mathbb{R}^n$ for which $f(x)$ is finite, the bit size of $f(x)$ is bounded by a polynomial in the size of x .*
- (ii) *We have $FP_{\mathbb{R}}^{\#P_{\mathbb{R}}} \subseteq FPAR_{\mathbb{R}}$. (To interpret this, represent ∞ by an element of $\mathbb{R} - \mathbb{N}$.)*

PROOF –

- (i) To prove the statement note that, given $x \in \mathbb{R}^n$, there exist polynomials p, q such that the set of witnesses for x is a semialgebraic subset of $\mathbb{R}^{p(n)}$ defined by a union of at most $2^{q(n)}$ basic semialgebraic sets, each of them described by a system of at most $q(n)$ inequalities of polynomials in $p(n)$ variables with degree at most $2^{q(n)}$. If this set is finite, its cardinality coincides with the number of its connected components. Now use the bounds of Corollary 5.1 on the number of connected components of basic semialgebraic sets.
- (ii) By the Theorem 6.7, to be presented later, it is sufficient to prove that $\#SAS_{\mathbb{R}}$ belongs to $FPAR_{\mathbb{R}}$. However, this follows directly from Theorem 4.3.

□

Remark 6.4 - A version of Theorem 6.6 holds over \mathbb{C} as well, with a proof similar to those over \mathbb{R} (use Theorem 4.5 instead of Theorem 4.3). It is immediate that the corresponding result also holds over $\{0, 1\}$.

6.4. Completeness for counting problems

Appropriate notions of reduction and completeness for counting problems are defined by extending these notions as we know them for decision problems.

Definition 6.5.

1. Let $f, g: \mathbb{K}^\infty \rightarrow \mathbb{N} \cup \{\infty\}$. We say that $\varphi: \mathbb{K}^\infty \rightarrow \mathbb{K}^\infty$ is a parsimonious reduction from f to g if φ can be computed in polynomial time and, for all $x \in \mathbb{K}^\infty$, $f(x) = g(\varphi(x))$.
2. We say that f Turing reduces to g if there exists an oracle machine which, with oracle g , computes f in polynomial time.
3. Let \mathcal{C} be $\#P_{\mathbb{K}}$ or $FP_{\mathbb{K}}^{\#P_{\mathbb{K}}}$. We say that a function g is hard for \mathcal{C} if, for every $f \in \mathcal{C}$, there is a parsimonious reduction from f to g . We say that g is \mathcal{C} -complete if, in addition, $g \in \mathcal{C}$.
4. The notions of Turing-hardness or Turing-completeness are defined similarly.

Valiant [114, 115] proved the $\#P$ -completeness (with respect to Turing reductions) of various enumeration and reliability problems. The most astonishing of his results is the $\#P$ -completeness of the problem to evaluate the permanent $\text{per}(A)$ of a matrix $A = [a_{i,j}] \in \{0, 1\}^{n \times n}$, defined by

$$\text{per}(A) := \sum_{\pi \in S_n} \prod_{i=1}^n a_{i, \pi(i)}.$$

This exhibited an unexpected difficulty for the computation of a function whose definition is only slightly different to that of the determinant, a problem known to be solvable in polynomial time. Important $\#P$ -hard problems are known in different areas, such as geometry (volume of polyhedras), knot theory (Jones polynomial), statistical physics (partition function), and network reliability, see [66, 116] for more information on this.

The version $\#P_{\mathbb{R}}$ over the reals was first considered by Meer in [83], but complete problems for it were not studied. Instead, the focus of Meer's paper are some logical properties of this class in terms of metafinite model theory.

Consider the following counting versions of the basic feasibility problems $\text{HN}_{\mathbb{C}}$, $\text{FEAS}_{\mathbb{R}}$, and $\text{SAS}_{\mathbb{R}}$.

$\#\text{HN}_{\mathbb{C}}$ (*Algebraic point counting*) Given a finite set of multivariate complex polynomials, count the number of complex common zeros, returning ∞ if this number is not finite.

$\#\text{FEAS}_{\mathbb{R}}$ (*Real algebraic point counting*) Given a multivariate real polynomial, count the number of its real roots, returning ∞ if this number is not finite.

$\#\text{SAS}_{\mathbb{R}}$ (*Semialgebraic point counting*) Given a semialgebraic set S , compute its cardinality if S is finite, and return ∞ otherwise.

As was to be expected, these counting problems turn out to be complete in the classes $\#\text{P}_{\mathbb{C}}$ and $\#\text{P}_{\mathbb{R}}$ respectively. The following was proved in [26].

Theorem 6.7.

1. The problem $\#\text{HN}_{\mathbb{C}}$ is $\#\text{P}_{\mathbb{C}}$ -complete.
2. The problems $\#\text{FEAS}_{\mathbb{R}}$ and $\#\text{SAS}_{\mathbb{R}}$ are $\#\text{P}_{\mathbb{R}}$ -complete with respect to Turing reductions.

Remark 6.5 - The version of $\text{SAS}_{\mathbb{R}}$ with semialgebraic sets given in conjunctive normal form is $\#\text{P}_{\mathbb{R}}$ -complete with respect to parsimonious reductions.

6.5. Boolean parts

It is common to restrict the input polynomials in the problems considered so far to polynomials with integer coefficients. The resulting problems can be encoded in a finite alphabet and studied in the classical Turing setting. In general, if L denotes a problem defined over \mathbb{R} or \mathbb{C} , we denote its restriction to integer inputs by $L^{\mathbb{Z}}$. This way, the discrete problems $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$, $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$, etc. are well defined.

The corresponding operation at the level of a complexity class is referred to as taking Boolean parts.

Definition 6.6. Let \mathcal{C} be a complexity class of decision problems over \mathbb{R} or \mathbb{C} . Its Boolean part is the classical complexity class

$$\text{BP}(\mathcal{C}) := \{S \cap \{0, 1\}^\infty \mid S \in \mathcal{C}\}.$$

Determining Boolean parts amounts to characterize, in terms of classical complexity classes, the power of resource bounded machines over \mathbb{R} or \mathbb{C} when their inputs are restricted to be binary. This has attracted quite some attention in real (or complex) complexity [23, 37, 38, 39, 69, 73].

Two of the most significant results concerning Boolean parts state that $\text{BP}(\text{P}_{\mathbb{C}}) \subseteq \text{P}^{\text{RP}}$ [38] and $\text{BP}(\text{PAR}_{\mathbb{R}}) = \text{PSPACE}/\text{poly}$ [37], and a third one is discussed in Proposition 6.2 below. For stating it, recall that RP denotes the classical complexity class of problems decidable by randomized machines in polynomial time with (one-sided) error. It is well-known that $\text{P}^{\text{RP}} \subseteq \Pi^2$, where Π^2 denotes a class in the second level of the polynomial hierarchy (see [1, 96] for details).

The following upper bound for $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ was obtained by Koiran [70].

Theorem 6.8. $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ belongs to RP^{NP} (and therefore to Π^2) under the generalized Riemann hypothesis GRH.

A natural restriction for real or complex machines (considered e.g. in [39, 69, 73]) is the requirement that no constants other than 0 and 1 appear in the machine program. Complexity classes arising by considering such constant-free machines are indicated by a superscript 0 as in $\text{P}_{\mathbb{R}}^0$, $\text{NP}_{\mathbb{R}}^0$, etc.

Theorem 6.8 provides an upper bound for $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$. On the other hand, the clear NP-hardness of $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ provides a lower bound. Yet there is a gap between NP and RP^{NP} and the problem of how to close it (with regard to $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$) is, as of today, an open question. The following result elaborates on that question.

Proposition 6.2.

- (i) $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ and $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$ are $\text{BP}(\text{NP}_{\mathbb{C}}^0)$ -complete.
- (ii) Assuming GRH, we have $\text{NP} \subseteq \text{BP}(\text{NP}_{\mathbb{C}}^0) \subseteq \text{RP}^{\text{NP}}$.

PROOF – The completeness of $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ in part (i) follows from the following fact. The $\text{FP}_{\mathbb{C}}$ -reduction from an arbitrary $\text{NP}_{\mathbb{C}}$ -problem to $\text{HN}_{\mathbb{C}}$ exhibited

in [16], when applied to a problem L in $\text{NP}_{\mathbb{C}}^0$, yields a FP-reduction from $L^{\mathbb{Z}}$ to $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$. This shows that $\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ is $\text{BP}(\text{NP}_{\mathbb{C}}^0)$ -complete. The completeness of $\text{DIM}_{\mathbb{C}}^{\mathbb{Z}}$ follows by examining the proof of Theorem 6.4 over \mathbb{C} , which is provided later (see Theorem 7.3).

For the reasoning above to hold it is essential that we only consider problems defined by $\text{NP}_{\mathbb{C}}$ -machines that do not use complex constants. Otherwise, these constants would appear as coefficients in the constructed polynomial system.

The second inclusion in part (ii) follows from part (i) and Theorem 6.8. The first inclusion is trivial. \square

It is believed [96, p. 255] that RP^{NP} has no complete problems. Thus, it follows from Proposition 6.2 that the equality $\text{BP}(\text{NP}_{\mathbb{C}}^0) = \text{RP}^{\text{NP}}$ is unlikely to hold.

6.6. Boolean parts of counting classes

Definition 6.6 can be extended to the classes $\#P_{\mathbb{C}}$ and $\#P_{\mathbb{R}}$ in an obvious way. The resulting classes, first defined and investigated in [26], will be of interest for us.

Definition 6.7. *The complexity class of geometric counting complex problems is defined as $\text{GCC} := \text{BP}(\#P_{\mathbb{C}}^0)$ and the class of geometric counting real problems as $\text{GCR} := \text{BP}(\#P_{\mathbb{R}}^0)$.*

These are classes of discrete counting problems, closed under parsimonious reductions, which can be located in a small region in the general landscape of classical complexity classes. Namely, we have

$$\#P \subseteq \text{GCC} \subseteq \text{GCR} \subseteq \text{FPSPACE},$$

where the rightmost inclusion follows from Theorem 6.6 and [37], or more directly from Theorem 4.3.

Proposition 6.3.

- (i) $\text{FEAS}_{\mathbb{R}}^{\mathbb{Z}}$, $\text{SAS}_{\mathbb{R}}^{\mathbb{Z}}$, and $\text{DIM}_{\mathbb{R}}^{\mathbb{Z}}$ are $\text{BP}(\text{NP}_{\mathbb{R}}^0)$ -Turing-complete.
- (ii) $\#\text{SAS}_{\mathbb{R}}^{\mathbb{Z}}$ and $\#\text{FEAS}_{\mathbb{R}}^{\mathbb{Z}}$ are GCR -Turing-complete.

(iii) $\#HN_{\mathbb{C}}^{\mathbb{Z}}$ is GCC-Turing-complete.

PROOF – For the hardness in part (i) we use the argument in the proof of Proposition 6.2(i), namely, that the reductions from an arbitrary $NP_{\mathbb{R}}$ -problem to $FEAS_{\mathbb{R}}$ or $SAS_{\mathbb{R}}$ yield reductions from problems in $BP(NP_{\mathbb{R}}^0)$ to $FEAS_{\mathbb{R}}^{\mathbb{Z}}$ or $SAS_{\mathbb{R}}^{\mathbb{Z}}$, respectively. For the hardness in part (ii) and (iii) one uses the reductions in the proof of Theorem 6.7. The memberships in all statements are clear except for $DIM_{\mathbb{R}}^{\mathbb{Z}}$, which follows by examining the proof of Theorem 6.4. \square

Remark 6.6 - One can show that $BP(NP_{\mathbb{C}}^0) = BP(NP_{\mathbb{C}})$ and $GCC = BP(\#P_{\mathbb{C}}^0) = BP(\#P_{\mathbb{C}})$. Hence it is immaterial whether we allow the use of complex machine constants in the definition of these classes or not. Moreover, it is possible to extend Proposition 6.2(ii) to $BP(FP_{\mathbb{C}}^{NPc}) \subseteq RP^{NP}$, assuming GRH. The proof relies on the possibility to eliminate complex constants using witness sequences, as developed in [15, 71, 74].

We can give some evidence that counting over \mathbb{C} is indeed harder than deciding feasibility over \mathbb{C} .

Corollary 6.1. *If $\#P_{\mathbb{C}} \subseteq FP_{\mathbb{C}}^{NPc}$, then the polynomial hierarchy would collapse at the second level, assuming GRH.*

PROOF – Assuming $\#P_{\mathbb{C}} \subseteq FP_{\mathbb{C}}^{NPc}$ and taking Boolean parts, we get by Remark 6.6

$$\#P \subseteq BP(\#P_{\mathbb{C}}) \subseteq BP(FP_{\mathbb{C}}^{NPc}) \subseteq RP^{NP} \subseteq \Pi^2.$$

Toda's theorem [113] states that $PH \subseteq P^{\#P}$. Hence we conclude $PH = \Pi^2$, which means that the polynomial hierarchy would collapse at the second level. \square

7. Structural complexity: Bézout, Euler, and Betti

In §5 we used the degree, Euler characteristic, and Betti numbers as yardsticks for measuring the complexity of a number of decision problems. In this section it is the problems of actually computing these quantities that become such yardsticks. Along with the change of focus from a quantity to the problem

of computing this quantity, we change the considered bounds from concrete to structural.

Unless otherwise stated, the results of this section are from [26].

7.1. Generic quantifiers

Several completeness results in the BSS-model rely on Koiran's method [72, 74, 75] to eliminate generic quantifiers in parametrized formulas. In the following we present this method as well as its extension developed in [26].

Recall that $\mathcal{F}_{\mathbb{R}}$ denotes the set of first order formulas over the language of the theory of ordered fields with constant symbols for real numbers. Similarly, let $\mathcal{F}_{\mathbb{C}}$ be the set of first order formulas over the language of the theory of fields with constant symbols for complex numbers.

Definition 7.1. *Let $F \in \mathcal{F}_{\mathbb{R}}$ have free variables a_1, \dots, a_k . We say that F is Zariski-generically true if the set of values $a \in \mathbb{R}^k$ not satisfying $F(a)$ has dimension strictly less than k . We express this fact by writing $\forall^* a F(a)$ using the generic universal quantifier \forall^* .*

The following lemma is easy to prove. Note that part (b) of it shows that $\forall^* a F(a)$ can be expressed by a first order formula. Hence by using the generic quantifier we still describe semialgebraic sets.

Lemma 7.1. *Let $F \in \mathcal{F}_{\mathbb{R}}$ have k free variables and coefficient field K , i.e., K is the field generated by the coefficients of all the polynomials occurring in F . Then $\forall^* a F(a)$ is equivalent to each of the following statements:*

- (a) $\{a \in \mathbb{R}^k \mid F(a)\}$ is dense in \mathbb{R}^k with respect to the Euclidean topology,
- (b) $\forall \epsilon \in \mathbb{R} \forall a \in \mathbb{R}^k \exists a' \in \mathbb{R}^k (\epsilon > 0 \Rightarrow F(a') \wedge \|a - a'\| < \epsilon)$,
- (c) $\forall a \in \mathbb{R}^k (a_1, \dots, a_k \text{ algebraically independent over } K \implies F(a))$.

Remark 7.1 - Let $F \in \mathcal{F}_{\mathbb{R}}$ have k free variables and coefficient field K . We say that F is *Euclidean-generically true*, written $\exists^* a F(a)$ using the generic existential quantifier \exists^* , iff $\neg \forall^* a \neg F(a)$. This is equivalent to each of the following statements:

- (a) $\{a \in \mathbb{R}^k \mid F(a)\}$ contains a nonempty open subset with respect to the Euclidean topology,
- (b) $\{a \in \mathbb{R}^k \mid F(a)\}$ has dimension k ,
- (c) there are $a_1, \dots, a_k \in \mathbb{R}$ algebraically independent over K such that $F(a)$ holds.

Remark 7.2 - Analogously, one can define \forall^* and \exists^* for formulas in $\mathcal{F}_{\mathbb{C}}$. It is not difficult to see, however, that these two quantifiers coincide over \mathbb{C} . Namely, Zariski genericity is the same as Euclidean genericity over \mathbb{C} .

Assume that $F(a) \in \mathcal{F}_{\mathbb{C}}$ has coefficient field K and the components of $\alpha \in \mathbb{C}^k$ are algebraically independent over K . Then $\forall^* a F(a)$ iff $F(\alpha)$ holds. Thus we may view α as a witness for the fact $\forall^* a F(a)$. We focus now on parametrized formulas $F(u, a)$ with parameter $u \in \mathbb{C}^p$ and look for witnesses which can be used for all values of the parameter u . This may not be attainable with a single witness point, but it turns out to be doable by using short sequences of witness points and taking a majority vote. In the sequel, $[n]$ denotes the set $\{1, \dots, n\}$.

Definition 7.2.

- (i) Let $F(u, a) \in \mathcal{F}_{\mathbb{C}}$ with free variables $u \in \mathbb{C}^p$ and $a \in \mathbb{C}^k$. We call a sequence $\alpha = (\alpha_1, \dots, \alpha_{4p+1}) \in (\mathbb{C}^k)^{4p+1}$ a witness sequence for F iff

$$\forall u \in \mathbb{C}^p \left(\forall^* a \in \mathbb{C}^k F(u, a) \iff |\{i \in [4p+1] \mid F(u, \alpha_i)\}| > 2p \right).$$

We denote the set of witness sequences of F by $W_{\mathbb{C}}(F)$.

- (ii) Let $F(u, a) \in \mathcal{F}_{\mathbb{R}}$ with free variables $u \in \mathbb{R}^{2p}$ and $a \in \mathbb{R}^k$. A sequence $\alpha = (\alpha_1, \dots, \alpha_{4p+1}) \in (\mathbb{R}^k)^{4p+1}$ is called a partial witness sequence for F iff

$$\forall u \in \mathbb{R}^{2p} \left(\forall^* a \in \mathbb{R}^k F(u, a) \implies |\{i \in [4p+1] \mid F(u, \alpha_i)\}| > 2p \right).$$

We denote the set of partial witnesses of F by $PW_{\mathbb{R}}(F)$.

Thus a witness sequence α can be used to certify that $\forall^* a \in \mathbb{C}^k F(u, a)$ holds by showing that the majority of the values α_i satisfies $F(u, \alpha_i)$. A

partial witness sequence cannot be used for that purpose, since in (ii) we only have an implication from left to right. However, partial witnesses will prove useful in situations where we a priori know that $\forall^* a \in \mathbb{R}^k F(u, a)$ holds.

We remark that [72] uses witness sequences over \mathbb{C} of length $2p+1$. We chose the length $4p+1$ just for uniformity of presentation. Witness sequences have been introduced in [72] for showing that $\text{DIM}_{\mathbb{C}}$ is $\text{NP}_{\mathbb{C}}$ -complete. We will discuss this completeness proof in §7.2. Partial witness sequences were introduced in [26] for proving the completeness of DEGREE and $\text{EULER}_{\mathbb{R}}^*$, defined in §7.2–§7.3.

The following lemma can be proved by a transcendence degree argument as for [72, Thm. 5.1].

Lemma 7.2.

- (i) $W_{\mathbb{C}}(F)$ is Zariski dense in $\mathbb{C}^{k(4p+1)}$ for any $F(u, a) \in \mathcal{F}_{\mathbb{C}}$.
- (ii) $PW_{\mathbb{R}}(F)$ is Zariski dense in $\mathbb{R}^{k(4p+1)}$ for any $F(u, a) \in \mathcal{F}_{\mathbb{R}}$.

Remark 7.3 - One could define a witness set $W_{\mathbb{R}}(F)$ for formulas $F(u, a) \in \mathcal{F}_{\mathbb{R}}$. However, this is not very useful, since $W_{\mathbb{R}}(F)$ does not need to be Zariski dense in $(\mathbb{R}^k)^{4p+1}$. For instance, let $p = 0, k = 1$ and consider the formula $F(a)$ expressing positivity of a . Then $W_{\mathbb{R}}(F) = \{\alpha \in \mathbb{R} \mid \alpha \leq 0\}$, which is not Zariski dense in \mathbb{R} .

The next theorem is similar to [75, Thm. 3].

Theorem 7.1. *Let $\mathbb{K} = \mathbb{R}$ (or $\mathbb{K} = \mathbb{C}$) and let $F(u, a) \in \mathcal{F}_{\mathbb{K}}$ be in prenex form with free variables $u \in \mathbb{R}^{2p}$ (or $u \in \mathbb{C}^p$) and $a \in \mathbb{K}^k$, n bounded variables, w alternating quantifier blocks, and m atomic predicates given by polynomials of degree at most $\delta \geq 2$ with integer coefficients of bit size at most ℓ .*

- (i) *In the case $\mathbb{K} = \mathbb{R}$ a partial witness sequence $\alpha \in PW_{\mathbb{R}}(F) \cap \mathbb{Z}^{k(4p+1)}$ can be computed by a straight-line program Γ of length $(kp)^{\mathcal{O}(1)} n^w \log(m\delta) + \mathcal{O}(\log \ell)$, which is division-free, has 1 as its only constant and no inputs.*
- (ii) *In the case $\mathbb{K} = \mathbb{C}$ a witness sequence $\alpha \in W_{\mathbb{C}}(F) \cap (\mathbb{Z} + i\mathbb{Z})^{k(4p+1)}$ can be computed by a straight-line program Γ of length $(kp)^{\mathcal{O}(1)} (2n)^w \log(m\delta) + \mathcal{O}(\log \ell)$, which is division-free, has 1, i as its only constants and no inputs.*

- (iii) In both cases, there exists a Turing machine which, on input $(p, k, n, w, m, \delta, \ell)$, computes Γ in time polynomial in the length of Γ . This machine does not depend on F .

A proof ingredient is the following easy lemma, whose proof can be found for instance in [71].

Lemma 7.3.

- (i) For positive integers k, L, D recursively define

$$\alpha_1 := 2^L, \alpha_j := 1 + \alpha_1(D + 1)^{j-1} \alpha_{j-1}^D \text{ for } 2 \leq j \leq k.$$

Then $h(\alpha_1, \dots, \alpha_k) \neq 0$ for any integer polynomial h in k variables of degree at most D and coefficients of absolute value less than 2^L .

- (ii) The sequence $\alpha_1, \dots, \alpha_k$ in part (i) can be computed by a straight-line program Γ performing $\mathcal{O}(k \log D + \log L)$ arithmetic operations and which has 1 as its only constant

Proof of Theorem 7.1

PROOF – (i) Assume first that $\mathbb{K} = \mathbb{R}$. We first replace the formula F by a quantifier free formula F' according to Theorem 4.1. Let M be the number of atomic predicates of F' , and D and L be upper bounds on the degree and the bit size of the occurring polynomials, respectively. We have

$$\log \max\{D, M\} \leq \mathcal{O}(kpn^w \log(m\delta)), \log L \leq \mathcal{O}(n^w \log(m\delta) + \log(2p + k + \ell)).$$

We replace the generic quantifier in Definition 7.2 according to Lemma 7.1(b) and thus write the formula as

$$\forall u \forall \epsilon \forall a \exists a' \left(\epsilon \leq 0 \vee (F'(u, a') \wedge \|a - a'\| < \epsilon) \implies \bigvee_I \bigwedge_{i \in I} F'(u, \alpha_i) \right),$$

where I runs over all $2p + 1$ -element subsets of $[4p + 1]$. This formula, let us call it ψ , defines $PW_{\mathbb{R}}(F)$ and is therefore Zariski-generically true by Lemma 7.2(ii). Note that ψ has $k(4p + 1)$ free variables and $2k + 2p + 1$ bounded variables, two quantifier blocks, and polynomials of degree at most D and bit size at most L . The number of atomic predicates of ψ equals $(4p + 2)M + 2$.

We again use Theorem 4.1 to replace the formula ψ by an equivalent quantifier free formula ψ' . Let M' be the upper bound on the number of atomic predicates and D' and L' denote the upper bounds on the degree and bit size of the polynomials h_{ij} occurring in ψ' , respectively, which are given by Theorem 4.1. Then we have

$$\log \max\{D', M'\} \leq (kp)^{\mathcal{O}(1)} n^w \log(m\delta)$$

and $\log L' \leq (kp)^{\mathcal{O}(1)} n^w \log(m\delta) + \mathcal{O}(\log \ell)$. We claim that

$$\bigcap_{i,j} \{\alpha \in \mathbb{R}^{k(4p+1)} \mid h_{ij}(\alpha) \neq 0\} \subseteq \{\alpha \in \mathbb{R}^{k(4p+1)} \mid \psi'(\alpha) \text{ holds}\} =: S.$$

Otherwise, there would be some $\alpha \notin S$ with $h_{ij}(\alpha) \neq 0$ for all i, j . Since the sign of h_{ij} does not change in some neighborhood U of α , U would be contained in the complement of S , which contradicts the fact that S is Zariski dense in $\mathbb{R}^{k(4p+1)}$.

According to Lemma 7.3, we can compute a point $\alpha \in \mathbb{Z}^{k(4p+1)}$ such that $h_{ij}(\alpha) \neq 0$ for all i, j by a straight-line program with $\mathcal{O}(kp \log D' + \log L')$ arithmetic operations. By plugging in the bounds on D' , L' the claim (i) follows.

(ii) For the case $\mathbb{K} = \mathbb{C}$ just represent a complex number by its real and imaginary part and argue as for part (i), using Lemma 7.2(i).

Claim (iii) is obvious. \square

Remark 7.4 - It follows from part (iii) of Theorem 7.1 that the element α in part (i) or part (ii) of this theorem can be computed by a machine over \mathbb{R} or \mathbb{C} , upon input $(p, k, n, w, m, \delta, \ell)$, in time order of the length of Γ . Note, however, that this computation may not be possible within these time bounds in the classical setting since the bit size of the components in α grows exponentially fast due to the repeated exponentiation (cf. Lemma 7.3).

7.2. Bézout

We are going to present the idea of the completeness proof from [26] of the following problem in the computational model of machines over \mathbb{C} .

DEGREE (Geometric degree) Given a finite set of complex polynomials, compute the geometric degree of its affine zero set.

Theorem 7.2. *The problem DEGREE is $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ -complete for Turing reductions.*

The difficult part of the proof is the upper bound, i.e., the membership of DEGREE to $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$. To show this membership, we have to describe a polynomial time algorithm over \mathbb{C} , which computes the degree using oracle calls to $\#\text{P}_{\mathbb{C}}$. The basic idea of our DEGREE algorithm is very simple. Let f_1, \dots, f_r be an instance for DEGREE and denote its zero set by Z . We first compute the dimension $d = \dim Z$ by calls to $\text{HN}_{\mathbb{C}}$ -oracles using Theorem 6.4. By definition, $\deg Z$ is the number of intersection points of Z with a generic affine subspace A of codimension d . If we could compute such an A , then the number of intersection points could be obtained by a call to $\#\text{HN}_{\mathbb{C}}$.

The difficulty is how to compute a generic affine subspace. Of course, the obvious way to turn this idea into an algorithm would be to choose the subspace A at random. This would yield a randomized algorithm for computing the degree. However, our goal is to choose A deterministically. We will do so using partial witness sequences for parametrized formulas as described in §7.1, for which we need to concisely express the degree.

In the following, we will parametrize affine subspaces of codimension d as follows. We denote by $A_a \subseteq \mathbb{C}^n$ the affine subspace of \mathbb{C}^n described by the system of linear equations $g_1(x) = 0, \dots, g_d(x) = 0$ with coefficient vector $a \in \mathbb{C}^k$, where $k = d(n + 1) = \mathcal{O}(n^2)$. Note that $\dim A_a \geq n - d$ for all a and $\forall^* a \dim A_a = n - d$. We have by the definition of degree

$$(7.1) \quad \forall^* a \in \mathbb{C}^k \quad |Z \cap A_a| = \deg Z.$$

It is clear that the above statement can be expressed by a first-order formula over \mathbb{C} . However, the obvious way to do this leads to a formula with exponentially many variables since $\deg Z$ can be exponentially large. Our goal is thus to express (7.1) in a more concise way. This will be achieved using the notion of transversality.

Let us first warm up by proving that $\text{DIM}_{\mathbb{C}}$ is $\text{NP}_{\mathbb{C}}$ -complete [72] (compare Theorem 6.4). Recall that $\text{DIM}_{\mathbb{C}}$ is the following problem: Given a finite set of

multivariate complex polynomials f_1, \dots, f_r with affine zero set $Z \subseteq \mathbb{C}^n$ and $d \in \mathbb{N}$, decide whether $\dim Z \geq d$.

Theorem 7.3. *The problem $\text{DIM}_{\mathbb{C}}$ is $\text{NP}_{\mathbb{C}}$ -complete*

PROOF – We will parametrize the system f_1, \dots, f_r by its vector of non-zero coefficients $u \in \mathbb{C}^p$, and we denote the corresponding zero set by Z_u . The $\text{NP}_{\mathbb{C}}$ -hardness of $\text{DIM}_{\mathbb{C}}$ follows by a trivial reduction from $\text{HN}_{\mathbb{C}}$ to $\text{DIM}_{\mathbb{C}}$.

To prove the upper bound, we use the following well-known characterization of the dimension

$$(7.2) \quad \dim Z_u \geq d \iff \forall^* a \ Z_u \cap A_a \neq \emptyset.$$

Let $F(u, a) \in \mathcal{F}_{\mathbb{C}}$ be the obvious parametrized formula expressing that $Z_u \cap A_a \neq \emptyset$. According to Theorem 7.1 and Remark 7.4 we can compute a witness sequence for $F(u, a)$ in $\text{FP}_{\mathbb{C}}$. The following nondeterministic polynomial time algorithm proves that $\text{DIM}_{\mathbb{C}}$ is contained in $\text{NP}_{\mathbb{C}}$.

```

input  $f_1, \dots, f_r$  with coefficient vector  $u$ 
compute a witness sequence  $\alpha = (\alpha_1, \dots, \alpha_{4p+1})$  of  $F(u, a)$ 
guess  $j_1, \dots, j_{2p+1} \in [1, \dots, 4p+1]$ 
guess  $x_1, \dots, x_{2p+1} \in \mathbb{C}^n$ 
for  $i = 1$  to  $2p+1$ 
    check whether  $x_i \in Z_u \cap A_{\alpha_{j_i}}$ 
if yes then accept else reject

```

Note that the algorithm accepts iff $|\{j \in [4p+1] \mid F(u, \alpha_j) \text{ holds}\}| > 2p$. Since α is a witness sequence for $F(u, a)$ this is equivalent to $\forall^* a \ F(u, a)$. But according to (7.2) this is equivalent to $\dim Z_u \geq d$. This shows correctness of the algorithm. \square

To continue with the problem DEGREE we need to define transversality. For this, we first recall the notion of a smooth point in a variety. To define smoothness we use Zariski tangent spaces.

Definition 7.3. *Let $Z \subseteq \mathbb{C}^n$ be an algebraic set, $x \in Z$, and f_1, \dots, f_r be generators of the vanishing ideal $\mathcal{I}(Z)$ of Z . The Zariski tangent space $T_x Z$ of*

Z at x is defined by

$$T_x Z = \mathcal{Z}(d_x f_1, \dots, d_x f_r)$$

where the differential of f at x , $d_x f: \mathbb{C}^n \rightarrow \mathbb{C}$, is the linear function defined by $d_x f X = \sum_{j=1}^n \partial_{X_j} f(x) X_j$. We say that x is a smooth point of Z if the dimension of $T_x Z$ equals the local dimension $\dim_x Z$ of Z at x . A point in Z which is not smooth is said to be a singular point of Z .

Note that $T_x Z$ is easy to compute from a set of generators of $\mathcal{I}(Z)$, but it may not be so, if instead we only have at hand an arbitrary set of polynomials with zero set Z .

Definition 7.4. Let $Z \subseteq \mathbb{C}^n$ be an algebraic set of dimension d and $A \subseteq \mathbb{C}^n$ be an affine subspace of codimension d .

1. A is called transversal to Z at $x \in Z \cap A$ iff x is a smooth point of Z and $T_x Z \oplus T_x A = \mathbb{C}^n$.
2. We say that A is transversal to Z when A is transversal to Z at all intersection points $x \in Z \cap A$ and if, additionally, there are no intersection points of Z and A at infinity. No intersection points at infinity means that $\overline{Z} \cap \overline{A} \subseteq \mathbb{C}^n$, where \overline{Z} and \overline{A} are the projective closures in $\mathbb{P}^n(\mathbb{C})$ of Z and A .

The following lemma shows that the transversality of A to Z can be used to certify that the number of intersection points of Z and A equals $\deg Z$. The proof can be essentially found in the book by Mumford [91, §5A], which is an excellent reference fitting well our geometric viewpoint.

Lemma 7.4. If $Z \subseteq \mathbb{C}^n$ is an algebraic set of dimension d and $k = d(n + 1)$, then we have:

- (i) $\forall^* a \in \mathbb{C}^k$ A_a is transversal to Z
- (ii) $\forall a \in \mathbb{C}^k$ (A_a is transversal to $Z \implies |Z \cap A_a| = \deg Z$).

Lemma 7.4 suggests to use transversality to concisely express degree. But, in turn, to express transversality a difficulty may arise. When we try to describe the Zariski tangent space of Z at a point x , the given equations $f_1 = 0, \dots, f_r =$

0 for Z might not generate the vanishing ideal of Z , since multiplicities might occur. In other words, the ideal generated by f_1, \dots, f_r might be different from the radical ideal, and it is not clear how to compute generators of the radical within the resources allowed. As a way out, we will express the tangent space and the transversality condition at x by a first order formula, in which all information regarding Z is given by a unary predicate expressing membership of points to Z .⁴

The following lemma is essential for the first order characterization we are seeking. We remark that the proof uses the notion of intersection multiplicity, which naturally appears in the context of Bézout's theorem.

Lemma 7.5. *Let $Z \subseteq \mathbb{C}^n$ be an algebraic set of dimension d and $A_a \subseteq \mathbb{C}^n$ be an affine subspace of codimension d , parametrized as above. For $x \in Z \cap A_a$ the following two conditions are equivalent:*

- (a) A_a is transversal to Z at x .
- (b) For every sufficiently small Euclidean neighborhood $U \subseteq \mathbb{C}^n$ of x there is a Euclidean neighborhood $V \subseteq \mathbb{C}^k$ of a such that for all $a' \in V$ the intersection $Z \cap A_{a'} \cap U$ contains exactly one point.

The following is an easy consequence of the characterization of transversality in Lemma 7.5. Again, we parametrize a system f_1, \dots, f_r of polynomials over \mathbb{C} by its vector of non-zero coefficients $u \in \mathbb{C}^p$, and we denote the corresponding zero set by Z_u .

Lemma 7.6. *For all $0 \leq d \leq n$ there is a first order formula $F_d(u, a)$ in $\mathcal{F}_{\mathbb{R}}$ in prenex form with seven quantifier blocks, $\mathcal{O}(n^2)$ bounded variables, and with $\mathcal{O}(p+n)$ atomic predicates given by integer polynomials of degree at most δ and bit size $\mathcal{O}(1)$, such that for all $u \in \mathbb{C}^p \simeq \mathbb{R}^{2p}$ with $\dim_{\mathbb{C}} Z_u = d$ and all $a \in \mathbb{C}^k$:*

$$F_d(u, a) \text{ is true} \iff A_a \text{ is transversal to } Z_u.$$

⁴This is closely related to the question of the expressive power of query languages for constraint spatial databases [78].

Proof of Theorem 7.2

PROOF – We begin with the membership of DEGREE to $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$. By Theorem 7.1(ii) and Remark 7.4, a partial witness sequence $\alpha = (\alpha_1, \dots, \alpha_{4p+1})$ for the formula $F_d(u, a)$ in Lemma 7.6 can be computed by a machine over \mathbb{C} , given input $(p, k, n, w, m, \delta, \ell)$, in time $(np)^{\mathcal{O}(1)} \log \delta$. Note that this quantity is polynomially bounded in the sparse input size $\mathcal{O}(np \log \delta)$.

We claim the correctness of the following algorithm for DEGREE.

```

input  $f_1, \dots, f_r$  with coefficient vector  $u$ 
compute  $d := \dim Z_u$  by oracle calls to  $\text{HN}_{\mathbb{C}}$  using Theorem 6.4
compute a partial witness sequence  $\alpha = (\alpha_1, \dots, \alpha_{4p+1})$  of  $F_d(u, a)$ 
for  $i = 1$  to  $4p + 1$ 
    compute  $N_i := |Z_u \cap A_{\alpha_i}|$  by an oracle call to  $\#\text{HN}_{\mathbb{C}}$ 
compute the majority  $N$  of the numbers  $N_1, \dots, N_{4p+1}$ 
return  $N$ 
    
```

Put $I := \{i \in [4p + 1] \mid F_d(u, \alpha_i) \text{ holds}\}$. Lemma 7.6 and Lemma 7.4(ii) imply that $N_i = \deg Z_u$ for all $i \in I$. Lemma 7.4(i) tells us that $\forall^* a F_d(u, a)$. Since α is a partial witness sequence, this implies that $|I| > 2p$ (cf. Definition (7.2)). This proves the claim.

It is obvious that the above algorithm can be implemented as a polynomial time oracle Turing machine over \mathbb{C} . This shows the membership.

To prove the hardness, note that, by Theorem 6.7, $\#\text{HN}_{\mathbb{C}}$ is $\#\text{P}_{\mathbb{C}}$ -complete. It is therefore sufficient to Turing reduce $\#\text{HN}_{\mathbb{C}}$ to DEGREE. The following reduction does so. For a given system of equations first decide whether its solution set Z is zero-dimensional by a call to $\text{HN}_{\mathbb{C}}$ using Theorem 6.4. This call to $\text{HN}_{\mathbb{C}}$ can be replaced by a call to DEGREE since $\text{HN}_{\mathbb{C}}$ reduces to DEGREE (recall $Z = \emptyset$ iff $\deg Z = 0$). If $\dim Z = 0$, then compute $N := \deg Z$ by a call to DEGREE and return N , otherwise return ∞ . \square

By combining the proof of Theorem 7.2 with Theorem 4.5 we get the following concrete upper bound on the complexity to compute the geometric degree. (Use that $p \leq r \binom{\delta+n}{n} \leq er\delta^n$.)

Corollary 7.1. *Given complex polynomials f_1, \dots, f_r of degree at most $\delta \geq 2$, one can compute the geometric degree of their zero set $Z \subseteq \mathbb{C}^n$ in parallel time*

$(n \log(r\delta))^{\mathcal{O}(1)}$ with a total number of $r^{\mathcal{O}(1)}\delta^{\mathcal{O}(n)}$ complex operations.

7.3. Euler-Poincaré

Our goal here is to sketch the proof of completeness from [26] of the following problem over \mathbb{R} .

EULER $_{\mathbb{R}}^*$ (Modified Euler characteristic) Given a semialgebraic set $S \subseteq \mathbb{R}^n$ as a union of basic semialgebraic sets

$$S = \bigcup_{i=1}^t \{x \in \mathbb{R}^n \mid g_i(x) = 0, f_{i1}(x) > 0, \dots, f_{ir_i}(x) > 0\},$$

decide whether S is empty and if not, compute $\chi^*(S)$.

Theorem 7.4. *The problem EULER $_{\mathbb{R}}^*$ is FP $_{\mathbb{R}}^{\#\text{P}}$ -complete with respect to Turing reductions.*

As for the problem DEGREE, the difficult part of the proof is the upper bound, that is, the membership of EULER $_{\mathbb{R}}^*$ to FP $_{\mathbb{R}}^{\#\text{P}}$. The basic idea is to use Morse theory in order to reduce the computation of the Euler characteristic to a counting problem. This can be implemented for smooth hypersurfaces with the help of the concept of partial witness sequences (Proposition 7.1). To finish, we show that the general case can be reduced to the case of a smooth hypersurface (Lemma 7.10).

We recall first some notions and facts from Morse theory. A general reference for this is [86].

Let Z be a differentiable manifold and $\varphi: Z \rightarrow \mathbb{R}$ be differentiable. A point $x \in Z$ is a *critical point* of φ if the differential $d_x\varphi: T_xZ \rightarrow \mathbb{R}$ vanishes. In this case, one may consider the *Hessian* $H_x\varphi: T_xZ \times T_xZ \rightarrow \mathbb{R}$ of φ at x , which is a symmetric bilinear form (defined by the second order derivatives of φ in local coordinates). The function φ is called *nondegenerate* at the critical point x if its Hessian is nondegenerate at x . The function φ is called a *Morse function* if all its critical points are nondegenerate.

We call the number of negative eigenvalues of a symmetric matrix or of a symmetric bilinear form its *index*. The *index* of φ at x is defined as the index of $H_x\varphi$. Throughout the paper, we will use the convenient notation $\{\varphi \leq r\} := \{x \in Z \mid \varphi(x) \leq r\}$.

The main theorem of Morse theory [86, Thm. 3.5] states the following.

Theorem 7.5. *Assume that $\varphi: Z \rightarrow \mathbb{R}$ is a Morse function on a differentiable manifold Z with finitely many critical points. Moreover, assume that $\{\varphi \leq r\}$ is compact for all $r \in \mathbb{R}$. Then Z has the homotopy type of a cell complex with one cell of dimension k for each critical point of φ of index k .*

We will use the following consequence of this result, adapted to the semialgebraic setting.

Corollary 7.2. *Let $Z \subseteq \mathbb{R}^n$ be a real algebraic manifold. Then,*

- (i) *The Euclidean distance function $L_a: Z \rightarrow \mathbb{R}, x \mapsto \|x - a\|^2$, is a Morse function for Zariski almost all $a \in \mathbb{R}^n$.*
- (ii) *Suppose that L_a is a Morse function on Z . Then the number N_k of critical points of L_a with index k is finite for all $0 \leq k \leq n$ and $\sum_{k=0}^n (-1)^k N_k$ equals the Euler characteristic $\chi(Z)$ of Z .*

Let \mathcal{H} be the set of polynomials $f \in \mathbb{R}[X_1, \dots, X_n]$ satisfying that $\mathcal{Z}(f) \neq \emptyset$ along with the regularity condition

$$(7.3) \quad \forall x \in \mathbb{R}^n (f(x) = 0 \Rightarrow \text{grad } f(x) \neq 0).$$

Note that $\mathcal{Z}(f)$ is a smooth hypersurface for $f \in \mathcal{H}$.

As in §7.2, we denote by $u \in \mathbb{R}^p$ the vector of non-zero coefficients of the polynomial $f = f_u$ of degree δ in X_1, \dots, X_n , and write $Z_u := \mathcal{Z}(f_u)$ for its zero set in \mathbb{R}^n .

The following lemma gives a certificate for L_a to be a Morse function on Z_u in the form of a parametrized first order formula. It plays a similar role for the completeness proof of $\text{EULER}_{\mathbb{R}}^*$ as the certificate for transversality for the completeness proof of DEGREE , which was provided in Lemma 7.6.

Lemma 7.7. *There is a first order formula $F(u, a)$ in $\mathcal{F}_{\mathbb{R}}$ in prenex form with one quantifier block, n bounded variables, and with $\mathcal{O}(n)$ atomic predicates given by integer polynomials of degree at most $\mathcal{O}(n\delta)$ and bit size $\mathcal{O}(n \log(np))$ such that, for all $u \in \mathbb{R}^p$ such that $f_u \in \mathcal{H}$ and all $a \in \mathbb{R}^n$, the following holds:*

$$F(u, a) \text{ is true} \iff L_a: Z_u \rightarrow \mathbb{R} \text{ is a Morse function.}$$

Consider the function $\chi_{\mathcal{H}} : \mathcal{H} \rightarrow \mathbb{Z}$, $f \mapsto \chi(\mathcal{Z}(f))$ computing the Euler characteristic of the smooth hypersurface $\mathcal{Z}(f)$ given by $f \in \mathcal{H}$. (Note that we do not consider the modified Euler characteristic here.)

Proposition 7.1. *The function $\chi_{\mathcal{H}}$ belongs to $\text{FP}_{\mathbb{R}}^{\#\text{P}_{\mathbb{R}}}$.*

PROOF – Let INDEX be the following decision problem. An input to INDEX is a tuple (u, a, x, k) , where u encodes a real polynomial f in n variables, $a, x \in \mathbb{R}^n$, and $k \in \mathbb{N}$. The question is to decide whether x is a critical point of index k of the function $L_a : Z_u \rightarrow \mathbb{R}$.

We claim that the problem INDEX is in $\text{P}_{\mathbb{R}}$. Indeed, given the tuple (u, a, x, k) , one first computes the Hessian $H_x L_a$, which can be explicitly expressed in terms of the first and second order partial derivatives of f . Then, one computes its characteristic polynomial (a computation known to be in $\text{FP}_{\mathbb{R}}$, see [16, 25]), and finally one uses Sturm’s algorithm to compute the number of real zeros in the interval $(-\infty, 0)$ (again in $\text{FP}_{\mathbb{R}}$, see [49]). Comparing this number with k decides INDEX for (u, a, x, k) .

Given (u, a) , let $\chi_+(u, a)$ denote the number of pairs (x, k) such that the tuple (u, a, x, k) is in INDEX and k is odd. Similarly, we define $\chi_-(u, a)$ by requiring that k is even. Since INDEX $\in \text{P}_{\mathbb{R}}$, the functions $\mathbb{R}^\infty \times \mathbb{R}^\infty \rightarrow \mathbb{N} \cup \{\infty\}$ mapping (u, a) to $\chi_+(u, a)$ and $\chi_-(u, a)$, respectively, are in $\#\text{P}_{\mathbb{R}}$.

We claim that

$$(7.4) \quad \chi(Z_u) = \chi_+(u, a) - \chi_-(u, a) \text{ if } L_a \text{ is a Morse function on } Z_u.$$

Indeed, if N_k denotes the number of critical points of L_a on Z_u of index k , we have

$$\chi(Z_u) = \sum_k (-1)^k N_k = \chi_+(u, a) - \chi_-(u, a)$$

by Corollary 7.2.

Lemma 7.7 and Theorem 7.1 imply that a partial witness sequence $\alpha \in (\mathbb{R}^n)^{2p+1}$ for the first order formula $F(u, a)$ certifying that $L_a : Z_u \rightarrow \mathbb{R}$ is a Morse function can be computed (uniformly) by a division-free straight-line program with $(np)^{\mathcal{O}(1)} \log \delta$ arithmetic operations. (Note that $u \in \mathbb{R}^p$, where p can be assumed to be even without loss of generality.)

The following algorithm computing $\chi_{\mathcal{H}}$ can be implemented as a polynomial time oracle machine querying oracles in $\#\mathbb{P}_{\mathbb{R}}$.

```

input  $f \in \mathcal{H}$  encoded by its coefficient vector  $u$ 
compute a partial witness sequence  $\alpha = (\alpha_1, \dots, \alpha_{2p+1})$  of  $F(u, a)$ 
for  $\ell = 1$  to  $2p + 1$ 
    compute  $\chi(u, \alpha_\ell) := \chi_+(u, \alpha_\ell) - \chi_-(u, \alpha_\ell)$ 
compute the majority  $\chi(u)$  of the numbers  $\chi(u, \alpha_1), \dots, \chi(u, \alpha_{2p+1})$ 
return  $\chi(u)$ 
    
```

In order to show that this algorithm actually computes the Euler characteristic of its input, put $\Lambda := \{\ell \in [2p + 1] \mid F(u, \alpha_\ell) \text{ holds}\}$. By definition of F we know that L_{α_ℓ} is a Morse function on Z_u for all $\ell \in \Lambda$. Hence, by (7.4), $\chi(Z_u) = \chi(u, \alpha_\ell)$ for all $\ell \in \Lambda$. On the other hand, by Proposition 7.2(i) we have $\forall^* a F(u, a)$. Since α is a partial witness sequence, this implies that $|\Lambda| > p$ (cf. Definition (7.2)). Therefore, the algorithm indeed computes the Euler characteristic of Z_u . \square

Lemma 7.10 below reduces the computation of the modified Euler characteristic of a real algebraic set to the computation of the (nonmodified) Euler characteristic of a *smooth* real hypersurface. Its proof uses a fact on covering maps and the following well-known (and easy to prove) result.

Lemma 7.8. *Let Z be a compact real algebraic n -dimensional manifold and $K \subseteq Z$ be a compact semialgebraic subset. Then*

$$\chi(Z - K) = \begin{cases} \chi(Z) - \chi(K) & \text{if } n \text{ is even,} \\ \chi(K) & \text{if } n \text{ is odd.} \end{cases}$$

A continuous map $p: X \rightarrow Y$ between topological spaces is called a *covering map* if there exists an open cover $\{U_\alpha\}$ of Y such that for each α , $p^{-1}(U_\alpha)$ is a disjoint union of open sets in X , each of which is mapped by p homeomorphically onto U_α (see e.g., [21, III.3]). If the cardinality of the fibre $p^{-1}(y)$ is constant for $y \in Y$, then this cardinality is called the *number of sheets* of the covering map. This condition is satisfied when Y is connected.

An example of a covering map with two sheets is the map $p: S^n \rightarrow \mathbb{P}^n(\mathbb{R})$, which identifies antipodal points. Note that $\chi(S^n) = 2\chi(\mathbb{P}^n(\mathbb{R}))$. This is no coincidence, as the following lemma shows.

Lemma 7.9. *If $X \rightarrow Y$ is a covering map with m sheets (m finite) and $\chi(Y)$ is defined, then $\chi(X) = m\chi(Y)$.*

For cell complexes, a proof of Lemma 7.9 can be found in [21, Prop. 13.5, p. 216]. For the more general case see for instance [106, p. 481].

Lemma 7.10. *Assume that $g \in \mathbb{R}[X_1, \dots, X_n]$ has even degree δ and put $G := X_0^{\delta+1}g(X_1/X_0, \dots, X_n/X_0)$. Then $\Phi := \mathcal{Z}(G - 1) \subseteq \mathbb{R}^{n+1}$ is a smooth affine hypersurface and we have*

$$\chi^*(\mathcal{Z}(g)) = \frac{(-1)^n}{2}(2 - \chi(\Phi)).$$

PROOF – Assume first that n is even. Put $Y := \mathcal{Z}(G) \subseteq \mathbb{P}^n(\mathbb{R})$ and consider the open subset $V := Y \cap \{X_0 \neq 0\}$ of Y , which is semialgebraically homeomorphic to $\mathcal{Z}(g)$. Since we homogenized with exponent $\delta + 1$, we have $Y - V = \mathcal{Z}_{\mathbb{P}^n(\mathbb{R})}(X_0) \simeq \mathbb{P}^{n-1}(\mathbb{R})$. By additivity of χ^* (Proposition 5.1) we have (cf. Example 2.1)

$$\chi^*(\mathcal{Z}(g)) = \chi^*(V) = \chi(Y) - \chi(\mathbb{P}^{n-1}(\mathbb{R})) = \chi(Y).$$

Note that 1 is a regular value of G , since $G = (\delta + 1)^{-1} \sum_i X_i \partial_{X_i} G$ by the homogeneity of G . Hence $\Phi = \{x \in \mathbb{R}^{n+1} \mid G(x) = 1\}$ is a smooth affine hypersurface. (We remark that over \mathbb{C} , Φ is called a Milnor fibre [41].) Put $U := \{x \in \mathbb{P}^n(\mathbb{R}) \mid G(x) \neq 0\}$. We claim that the canonical map

$$\pi: \Phi \rightarrow U, (x_0, \dots, x_n) \mapsto (x_0 : \dots : x_n)$$

is a covering map with two sheets. Indeed, $\pi^{-1}(U \cap \{X_i \neq 0\}) = (\Phi \cap \{X_i > 0\}) \cup (\Phi \cap \{X_i < 0\})$, and π induces homeomorphisms from both $\Phi \cap \{X_i > 0\}$ and $\Phi \cap \{X_i < 0\}$ to $U \cap \{X_i \neq 0\}$, respectively (δ is even).

By Lemma 7.9 we have $\chi(\Phi) = 2\chi(U)$. On the other hand, by Lemma 7.8 we get $\chi(U) = \chi(\mathbb{P}^n(\mathbb{R})) - \chi(Y) = 1 - \chi(Y)$. Altogether, we obtain

$$\chi^*(\mathcal{Z}(g)) = \chi(Y) = 1 - \chi(U) = 1 - \frac{1}{2}\chi(\Phi).$$

The case where n is odd can be treated similarly. □

Proof of Theorem 7.4

PROOF – (Sketch) Assume we are given a real algebraic input set $\mathcal{Z}(g)$. Lemma 7.10 reduces the computation of $\chi^*(\mathcal{Z}(g))$ to the computation of the Euler characteristic $\chi(\Phi)$ of the smooth affine real hypersurface Φ . Proposition 7.1 shows that $\chi(\Phi)$ can be computed in $\text{FP}_{\mathbb{R}}^{\#\mathbb{P}_{\mathbb{R}}}$. The extension to basic semialgebraic input sets is straightforward. To treat a union of basic semialgebraic sets, one can use the inclusion-exclusion principle, taking into account that χ^* behaves additively with respect to disjoint unions. We omit the details. This finishes our sketch of the proof of the upper bound.

For the lower bound recall that by Theorem 6.7, $\#\text{FEAS}_{\mathbb{R}}$ is $\#\mathbb{P}_{\mathbb{R}}$ -complete. To prove the Turing-hardness of $\text{EULER}_{\mathbb{R}}^*$ for $\#\mathbb{P}_{\mathbb{R}}$, it is therefore sufficient to Turing reduce $\#\text{FEAS}_{\mathbb{R}}$ to $\text{EULER}_{\mathbb{R}}^*$. The following reduction does so. For a given real polynomial first decide whether its solution set Z is zero-dimensional by a call to $\text{FEAS}_{\mathbb{R}}$ using Theorem 6.4. This call to $\text{FEAS}_{\mathbb{R}}$ can be replaced by a call to $\text{EULER}_{\mathbb{R}}^*$ since $\text{FEAS}_{\mathbb{R}}$ reduces to $\text{EULER}_{\mathbb{R}}^*$ (this follows from the case distinction in the definition of the problem $\text{EULER}_{\mathbb{R}}^*$). If $\dim Z = 0$, then compute $N := \chi^*(Z)$ by a call to $\text{EULER}_{\mathbb{R}}^*$ and return N , otherwise return ∞ . \square

7.4. Bézout, Euler-Poincaré, and Betti in the Turing model

For the Turing model, we can deduce completeness results for the discrete versions $\text{DEGREE}^{\mathbb{Z}}$ and $\text{EULER}_{\mathbb{R}}^{*\mathbb{Z}}$ of the problems to compute the degree and the modified Euler characteristic, respectively. The completeness results are in terms of the complexity classes GCC and GCR introduced in §6.6. It is also possible to obtain a completeness result for the problem $\text{EULER}_{\mathbb{R}}$ of computing the (non-modified) Euler characteristic, defined as follows:

$\text{EULER}_{\mathbb{R}}$ (*Euler characteristic for basic semialgebraic sets*) Given a basic semialgebraic set $S = \{x \in \mathbb{R}^n \mid g(x) = 0, f_1(x) > 0, \dots, f_r(x) > 0\}$, decide whether S is empty and if not, compute $\chi(S)$.

Theorem 7.6.

- (i) $\text{DEGREE}^{\mathbb{Z}}$ is FP^{GCC} -complete with respect to Turing reductions.

(ii) $\text{EULER}_{\mathbb{R}}^{\mathbb{Z}}$ and $\text{EULER}_{\mathbb{R}}^{*\mathbb{Z}}$ are FP^{GCR} -complete with respect to Turing reductions.

PROOF – (i) The proof given in §7.2 for the membership of DEGREE to $\text{FP}_{\mathbb{C}}^{\#\text{P}_{\mathbb{C}}}$ applies in our case with only one modification. The algorithm in the proof of Theorem 7.2 computes the partial witness sequence α (this is done in $\text{FP}_{\mathbb{C}}$) and then performs $4p + 1$ oracle calls to $\#\text{P}_{\mathbb{C}}$ to obtain the numbers N_i for $i \in [4p + 1]$. While it is clear that the computation of α is in $\text{BP}(\text{FP}_{\mathbb{C}})$, it is equally clear that it is not in $\text{FP}_{\mathbb{C}}$ due to the exponential coefficient growth caused by repeated powering (cf. Lemma 7.3). A way to solve this is to “move” the computation of α to the query. That is, one considers the problem of computing N_i with input (u, i) . Clearly, this problem is in $\text{BP}(\#\text{P}_{\mathbb{C}})$: one first computes α in $\text{FP}_{\mathbb{C}}$ and then N_i in $\#\text{P}_{\mathbb{C}}$.

The hardness of $\text{DEGREE}^{\mathbb{Z}}$ follows as for Theorem 7.2 using Proposition 6.3 and Proposition 6.2(i).

(ii) The proof for $\text{EULER}_{\mathbb{R}}^{*\mathbb{Z}}$ is a modification of the proof of Theorem 7.4, similar as for part (i).

(iii) The upper bound for $\text{EULER}_{\mathbb{R}}^{\mathbb{Z}}$ requires some additional reasoning, which is omitted here. \square

Remark 7.5 - The algorithms for $\text{DEGREE}^{\mathbb{Z}}$ and $\text{EULER}_{\mathbb{R}}^{*\mathbb{Z}}$ above can be further simplified. Since we can bound the description size of the formula $F(u, a)$ by taking into account a bound on the bit size of the given $u \in \mathbb{Z}^p$, the input vector u does not need to be considered as a parameter any more. Therefore, we may take $p = 0$. The partial witness sequence then consists of a single vector $\alpha \in \mathbb{Z}^k$ and only one oracle call to $\#\text{HN}_{\mathbb{C}}^{\mathbb{Z}}$ (or two oracle calls to $\#\text{FEAS}_{\mathbb{R}}^{\mathbb{Z}}$) are needed.

Finally, we briefly discuss the following problems ($k \in \mathbb{N}$):

$\#\text{CC}_{\mathbb{R}}$ (*Counting connected components*) Given a semialgebraic set S , compute the number of its connected components.

$\text{BETTI}(k)_{\mathbb{R}}$ (*kth Betti number of a real algebraic set*) Given a real multivariate polynomial, compute the k th Betti number of its real zero set.

BM-BETTI(k) $_{\mathbb{R}}$ (*k*th Borel-Moore Betti number of a real algebraic set) Given a real multivariate polynomial, compute the *k*th Borel-Moore Betti number of its real zero set.

For the problems related to Betti numbers, we restrict the input to be a real algebraic set. Since we will only talk about lower bounds for these problems, this restriction makes our results stronger. Note that $\text{BETTI}(0)_{\mathbb{R}}$ is just the restriction of $\#\text{CC}_{\mathbb{R}}$ to real algebraic sets.

We will focus here on the discretized versions of the above problems, where the input polynomials have integer coefficients, and study these in the Turing model.

The following upper bound was first shown by Canny [30]. It follows immediately from Theorem 4.4.

Theorem 7.7. *The problem $\#\text{CC}_{\mathbb{R}}^{\mathbb{Z}}$ is in FPSPACE.*

From a result by Reif [99, 100] on the PSPACE-hardness of a generalized movers problem in robotics, it follows easily that the problem $\#\text{CC}_{\mathbb{R}}^{\mathbb{Z}}$ is in fact FPSPACE-complete. In [26] an alternative proof of the FPSPACE-hardness of this problem was given. This alternative proof, based on ideas exposed in [27], sharpens Reif’s lower bound for $\#\text{CC}_{\mathbb{R}}^{\mathbb{Z}}$ since it shows that $\#\text{CC}_{\mathbb{R}}^{\mathbb{Z}}$ remains FPSPACE-hard when restricted to compact real algebraic sets. Using this sharpened result the following was proved in [26].

Theorem 7.8. *For any $k \in \mathbb{N}$ both problems $\text{BETTI}(k)_{\mathbb{R}}^{\mathbb{Z}}$ and $\text{BM-BETTI}(k)_{\mathbb{R}}^{\mathbb{Z}}$ are FPSPACE-hard with respect to Turing reductions.*

8. Open problems

Problem 8.1. *Can one characterize GCR or GCC in terms of known classical complexity classes?*

Problem 8.2. *Toda’s theorem [113] states that $\text{PH} \subseteq \text{FP}^{\#\text{P}}$. Is there an analogue of this over \mathbb{R} or over \mathbb{C} ?*

Problem 8.3. *It is known that the problem to count the number of connected components of a semialgebraic set is in $\text{FPAR}_{\mathbb{R}}$. Is it hard in this class? We know that the corresponding result is true in the additive setting [27].*

Problem 8.4. *Can Betti numbers of semialgebraic sets be computed in $\text{FPAR}_{\mathbb{R}}$? We know that, in the additive setting, the computation of Betti numbers of semi-linear sets is FPAR_{add} -complete [27].*

Problem 8.5. *What is the complexity to check irreducibility of algebraic varieties over \mathbb{C} ? And what is the complexity of counting the number of irreducible components of algebraic varieties?*

References

- [1] BALCÁZAR, J.L., DÍAZ, J., and GABARRÓ, J.: *Structural complexity I*, Springer Verlag (1988).
- [2] BARVINOK, A.I.: On the Betti numbers of semialgebraic sets defined by few quadratic inequalities, *Math. Zeit.* **225** (1997), 231–244.
- [3] BASU, S.: Computing the Betti numbers of arrangements via spectral sequences, *J. Comp. Syst. Sci.*, to appear.
- [4] BASU, S.: On bounding the Betti numbers and computing the Euler characteristic of semi-algebraic sets, *Discrete Comput. Geom.* **22** (1) (1999), 1–18.
- [5] BASU, S.: Different bounds on the different Betti numbers of semi-algebraic sets, *Discrete Comput. Geom.* **30** (2003), 65–85.
- [6] BASU, S., POLLACK, R., and ROY, M.-F.: On the combinatorial and algebraic complexity of quantifier elimination, *J. ACM* **43** (1996), 1002–1045.
- [7] BASU, S., POLLACK, R., and ROY, M.-F.: Computing roadmaps of semi-algebraic sets on a variety, *J. Amer. Math. Soc.* **13** (1999), 55–82.
- [8] BASU, S., POLLACK, R., and ROY, M.-F.: *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics **10**, Springer Verlag (2003).
- [9] BAUR, W. and STRASSEN, V.: The complexity of partial derivatives, *Theoret. Comp. Sci.* **22** (1983), 317–330.
- [10] BEN-OR, M.: Lower bounds for algebraic computation trees, *Proc. 15th ACM STOC*, Boston (1983), 80–86.
- [11] BENEDETTI, R. and RISLER, J.-J.: *Real algebraic and semi-algebraic sets*, Hermann (1990).
- [12] BJÖRNER, A.: Subspace arrangements, *Proc. of 1st European Congress of Mathematics (Paris, 1992)*, Birkhäuser (1992), 321–370.

- [13] BJÖRNER, A. and LOVÁSZ, L.: Linear decision trees, subspace arrangements and Möbius functions, *J. Amer. Math. Soc.* **7** (3) (1994), 677–706.
- [14] BJÖRNER, A., LOVÁSZ, L., and YAO, A.C.: Linear decision trees: volume estimates and topological bounds, *Proc. 24th ACM STOC* (1992), 171–177.
- [15] BLUM, L., CUCKER, F., SHUB, M., and SMALE, S.: Algebraic Settings for the Problem “ $P \neq NP$?”, *The mathematics of numerical analysis, Lectures in Applied Mathematics* **32**, Amer. Math. Soc. (1996), 125–144.
- [16] BLUM, L., CUCKER, F., SHUB, M., and SMALE, S.: *Complexity and Real Computation*, Springer (1998).
- [17] BLUM, L., SHUB, M., and SMALE, S.: On a theory of computation and complexity over the real numbers, *Bull. Amer. Math. Soc.* **21** (1989), 1–46.
- [18] BOCHNAK, J., COSTE, M., and ROY, M.F.: Géométrie algébrique réelle, *Ergebnisse der Mathematik und ihrer Grenzgebiete*, 3. Folge **12**, Springer Verlag (1987).
- [19] BOREL, A. and MOORE, J.C.: Homology theory for locally compact spaces, *Michigan Math. J.* **7** (1960), 137–159.
- [20] BORODIN, A.B.: On relating time and space to size and depth, *SIAM J. Comp.* **6** (1977), 733–744.
- [21] BREDON, G.E.: Topology and geometry, *GTM* **139**, Springer Verlag (1993).
- [22] BÜRGISSER, P.: On the parallel complexity of the polynomial ideal membership problem, *J. Compl.* **14** (1998), 176–189.
- [23] BÜRGISSER, P.: Cook’s versus Valiant’s hypothesis, *Theoret. Comp. Sci.* **235** (2000), 71–88.
- [24] BÜRGISSER, P.: Lower bounds and real algebraic geometry, Algorithmic and Quantitative Real Algebraic Geometry (S. Basu and L. Gonzales-Vega, eds.), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **60**, AMS (2003), 35–54.
- [25] BÜRGISSER, P., CLAUSEN, M., and SHOKROLLAHI, M.A.: *Algebraic complexity theory*, Grundlehren der mathematischen Wissenschaften, vol. 315, Springer Verlag (1997).
- [26] BÜRGISSER, P. and CUCKER, F.: Counting complexity classes for numeric computations II: Algebraic and semialgebraic sets, <http://www.arxiv.org/abs/cs/cs.CC/0312007>. Submitted.
- [27] BÜRGISSER, P. and CUCKER, F.: Counting complexity classes for numeric computations I: Semilinear sets, *SIAM J. Comp.* (2004), to appear.

- [28] BÜRGISSER, P., LICKTEIG, T., and SHUB, M.: Test complexity of generic polynomials, *J. Compl.* **8** (1992), 203–215.
- [29] CANNY, J.: *The complexity of robot motion planning*, ACM Doctoral Dissertation Awards, vol. 1987, MIT Press, Cambridge, MA (1988).
- [30] CANNY, J.: Some algebraic and geometric computations in PSPACE, *Proc. 20th Ann. ACM STOC* (1988), 460–467.
- [31] CANNY, J.: Computing roadmaps of general semi-algebraic sets, *The Computer Journal* **36** (5) (1993), 504–514.
- [32] CHISTOV, A.L.: Polynomial-time computation of the dimension of algebraic varieties in zero-characteristic, *J. Symb. Comp.* **22** (1996), 1–25.
- [33] COLLINS, G.E.: *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, LNCS, vol. 33, Springer-Verlag (1975), 134–183.
- [34] COOK, S.A.: The complexity of theorem proving procedures, *Proc. 3rd ACM STOC* (1971), 151–158.
- [35] CUCKER, F.: $P_{\mathbb{R}} \neq NC_{\mathbb{R}}$, *J. Compl.* **8** (1992), 230–238.
- [36] CUCKER, F.: Real computations with fake numbers, *J. Compl.* **18** (2002), 104–134.
- [37] CUCKER, F. and GRIGORIEV, D.YU.: On the power of real Turing machines over binary inputs, *SIAM J. Comp.* **26** (1997), 243–254.
- [38] CUCKER, F., KARPINSKI, M., KOIRAN, P., LICKTEIG, T., and WERTHER, K.: On real Turing machines that toss coins, *Proc. 27th ACM STOC*, Las Vegas (1995), 335–342.
- [39] CUCKER, F. and KOIRAN, P.: Computing over the reals with addition and order: Higher complexity classes, *J. Compl.* **11** (1995), 358–376.
- [40] DAVENPORT, J.H. and HEINTZ, J.: Real quantifier elimination is doubly exponential, *J. Symb. Comp.* **5** (1988), 29–35.
- [41] DIMCA, A.: *Singularities and topology of hypersurfaces*, Universitext, Springer Verlag (1992).
- [42] DOBKIN, D. and LIPTON, R.J.: A lower bound of $\frac{1}{2}n^2$ on linear search programs for the knapsack problem, *J. Comp. Syst. Sci.* **16** (1978), 413–417.
- [43] FISCHER, M.J. and RABIN, M.: *Super-exponential complexity of Presburger arithmetic*, Complexity of Computation (R.M. Karp, ed.), Amer. Math. Soc., Providence, RI (1974), 27–41.

- [44] FITCHAS, N., GALLIGO, A., and MORGENSTERN, J.: Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields, *J. Pure Appl. Alg.* **67** (1990), 1–14.
- [45] FULTON, W.: *Introduction to toric varieties*, Annals of Math. Studies, Princeton University Press (1993).
- [46] GABRIELOV, A., VOROBYOV, N., and ZELL, T.: Betti numbers of semialgebraic and sub-Pfaffian sets, *J. London Math. Soc.* **69** (2004), 27–43.
- [47] GAREY, M.R. and JOHNSON, D.S.: *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman and Company, New York (1979).
- [48] VON ZUR GATHEN, J.: Parallel arithmetic computations: a survey, *Proc. 12th Symp. Math. Found. Comput. Sci.*, LNCS **233**, Bratislava (1986), 93–112.
- [49] VON ZUR GATHEN, J. and GERHARD, J.: *Modern computer algebra*, University Press, Cambridge (1999).
- [50] GIUSTI, M. and HEINTZ, J.: La détermination des points isolés et de la dimension d’une variété algébrique peut se faire en temps polynomial, *Proc. Cortona Conference on Computational Algebraic Geometry and Commutative Algebra* (D. Eisenbud and L. Robbiano, eds.), Cambridge University Press (1993).
- [51] GIUSTI, M., HEINTZ, J., and SABIA, J.: On the efficiency of effective Nullstellensätze, *Comput. Complexity* **3** (1) (1993), 56–95.
- [52] GORESKEY, M. and MACPHERSON, R.D.: Stratified Morse theory, *Ergebnisse der Mathematik und ihrer Grenzgebiete*, 3. Folge **14**, Springer Verlag (1988).
- [53] GRIGORIEV, D.YU.: Complexity of deciding Tarski algebra, *J. Symb. Comp.* **5** (1988), 65–108.
- [54] GRIGORIEV, D.YU., CANNY, J., and VOROBYOV, N.: Finding connected components of a semialgebraic set in subexponential time, *AAECC* **2** (1992), 217–238.
- [55] GRIGORIEV, D.YU. and VOROBYOV, N.: Solving systems of polynomial inequalities in subexponential time, *J. Symb. Comp.* **5** (1988), 37–64.
- [56] GRIGORIEV, D.YU. and VOROBYOV, N.: Counting connected components of a semialgebraic set in subexponential time, *Comp. Compl.* **2** (2) (1992), 133–186.
- [57] HARRIS, J.: *Algebraic geometry: A first course*, GTM, Springer Verlag, New York (1992).
- [58] HARTSHORNE, R.: *Algebraic geometry*, GTM, Springer Verlag (1977).

- [59] HATCHER, A.: *Algebraic topology*, Cambridge University Press, Cambridge (2002).
- [60] HEINTZ, J.: Definability and fast quantifier elimination in algebraically closed fields, *Theoret. Comp. Sci.* **24** (1983), 239–277.
- [61] HEINTZ, J. and MORGENSTERN, J.: On the intrinsic complexity of elimination theory, *Journal of Complexity* **9** (1993), 471–498.
- [62] HEINTZ, J., ROY, M.F., and SOLERNÓ, P.: Sur la complexité du principe de Tarski-Seidenberg, *Bull. Soc. Math. France* **118** (1990), 101–126.
- [63] HEINTZ, J., ROY, M.F., and SOLERNÓ, P.: Description of the connected components of a semialgebraic set in single exponential time, *Discrete Comput. Geom.* **11** (2) (1994), 121–140.
- [64] HIRONAKA, H.: Triangulation of algebraic sets, *Proceedings of Symposia in Pure Mathematics* **29**, Amer. Math. Soc. (1975), 165–185.
- [65] HIRZEBRUCH, F.: , *New topological methods in algebraic geometry*, Springer Verlag (1965).
- [66] JERRUM, M.: *Counting, sampling and integrating: Algorithms and complexity*, Birkhäuser, Basel (2003).
- [67] KARP, R.M.: *Reducibility among combinatorial problems*, Complexity of Computer Computations (R.E. Miller and J.W. Thatcher, eds.), New York (1972), 85–104.
- [68] KNUTH, D.E.: The analysis of algorithms, *Actes du congrès international des Mathématiciens* **3**, Nice (1970), 269–274.
- [69] KOIRAN, P.: Computing over the reals with addition and order, *Theoret. Comp. Sci.* **133** (1994), 35–47.
- [70] KOIRAN, P.: Hilbert’s Nullstellensatz is in the polynomial hierarchy, *J. Compl.* **12** (1996), 273–286.
- [71] KOIRAN, P.: Elimination of constants from machines over algebraically closed fields, *J. Compl.* **13** (1997), 65–82.
- [72] KOIRAN, P.: Randomized and deterministic algorithms for the dimension of algebraic varieties, *Proc. 38th FOCS* (1997), 36–45.
- [73] KOIRAN, P.: A weak version of the Blum, Shub & Smale model, *J. Comp. Syst. Sci.* **54** (1997), 177–189.
- [74] KOIRAN, P.: Elimination of parameters in the polynomial hierarchy, *Theoret. Comp. Sci.* **215** (1999), 289–304.

- [75] KOIRAN, P.: The real dimension problem is $\text{NP}_{\mathbf{R}}$ -complete, *J. Compl.* **15** (2) (1999), 227–238.
- [76] KOIRAN, P.: Circuits versus trees in algebraic complexity, *Proc. STACS 2000*, LNCS **1770**, Springer Verlag (2000), 35–52.
- [77] KUNG, H.T.: New algorithms and lower bounds for the parallel evaluation of certain rational expressions and recurrences, *J. ACM* **23** (1976), 252–261.
- [78] KUPER, G.M., PAREDAENS, J., and LIBKIN, L.: *Constraint databases*, Springer-Verlag (2000).
- [79] LAKATOS, I.: *Proofs and refutations: the logic of mathematical discovery*, Cambridge University Press (1976).
- [80] LEHMER, D.H.: Euclid’s algorithm for large numbers, *Amer. Math. Monthly* **45** (1938), 227–233.
- [81] LEVIN, L.A.: Universal search problems, *Problems of Information Transmission* **9** (1973), 265–266.
- [82] LICKTEIG, T.: On semialgebraic decision complexity, Tech. Report TR-90-052, *Int. Comp. Sc. Inst.*, Habilitationsschrift, Universität Tübingen, Berkeley (1990).
- [83] MEER, K.: Counting problems over the reals, *Theoret. Comp. Sci.* **242** (2000), 41–58.
- [84] MEISER, S.: Point location in arrangements of hyperplanes, *Information and Computation* **106** (1993), 286–303.
- [85] MEYER AUF DER HEIDE, F.: A polynomial linear search algorithm for the n -dimensional knapsack problem, *J. ACM* **31** (1984), 668–676.
- [86] MILNOR, J.: Morse theory, *Annals of Math. Studies* **51**, Princeton University Press (1963).
- [87] MILNOR, J.: On the Betti numbers of real varieties, *Proc. AMS* **15** (1964), 275–280.
- [88] MOENCK, R.T.: Fast computation of GCDs, *Proc. 5th ACM STOC* (1973), 142–151.
- [89] MONTAÑA, J.L., MORAIS, J.E., and PARDO, L.M.: Lower bounds for arithmetic networks II: Sum of Betti numbers, *AAECC* **7** (1996), 41–51.
- [90] MONTAÑA, J.L. and PARDO, L.M.: Lower bounds for arithmetic networks, *AAECC* **4** (1993), 1–24.
- [91] MUMFORD, D.: *Algebraic geometry I: Complex projective varieties*, Springer Verlag (1976).

- [92] MUNKRES, J.R.: *Elements of algebraic topology*, Addison-Wesley Publishing Company, Menlo Park, CA (1984).
- [93] NUESKEN, M.: *Topologische Grenzen für Algebraische Berechnungsbäume*, unpublished report, Universität Konstanz (1998).
- [94] OLEĀNIK, O.A.: Estimates of the Betti numbers of real algebraic hypersurfaces, *Math. Sb. (N.S.)* **28** (70) (1951), 635–640.
- [95] OLEĀNIK, O.A. and PETROVSKII, I.B.: On the topology of real algebraic surfaces, *Izv. Akad. Nauk SSSR* **13** (1949), 389–402.
- [96] PAPANITRIOU, C.H.: *Computational complexity*, Addison-Wesley (1994).
- [97] POIZAT, B.: Les petits cailloux, *Nur Al-Mantiq War-Ma'rifah* **3**, Aléas, Lyon (1995).
- [98] PREPARATA, F.P. and SHAMOS, M.I.: *Computational geometry, an introduction*, Texts and Monographs in Computer Science, Springer Verlag (1985).
- [99] REIF, J.H.: Complexity of the mover's problem and generalizations, *Proc. 20th FOCS* (1979), 421–427.
- [100] REIF, J.H.: *Complexity of the generalized mover's problem*, Planning, Geometry and Complexity of Robot Motion (J.T. Schwartz, M. Sharir, and J. Hopcroft, eds.), Ablex Publishing Corporation (1987), 267–281.
- [101] RENEGAR, J.: On the computational complexity and geometry of the first-order theory of the reals. part I, II, III, *J. Symb. Comp.* **13** (3) (1992), 255–352.
- [102] ROTMAN, J.J.: *An introduction to algebraic topology*, GTM **119**, Springer Verlag (1988).
- [103] SCHÖNHAGE, A.: Schnelle Berechnung von Kettenbruchentwicklungen, *Act. Inf.* **1** (1971), 139–144.
- [104] SCHWARTZ, J.T. and SHARIR, M.: On the “piano movers” problem. II. General techniques for computing topological properties of real algebraic manifolds, *Adv. in Appl. Math.* **4** (3) (1983), 298–351.
- [105] SHAFAREVICH, I.R.: *Basic algebraic geometry*, Springer Verlag (1974).
- [106] SPANIER, E.: *Algebraic topology*, MacGraw-Hill (1966).
- [107] STEELE, J.M. and YAO, A.C.: Lower bounds of algebraic decision trees, *J. Algorithms* **3** (1982), 1–8.
- [108] STEENROD, N.: *Topology of fibre bundles*, Princeton University Press (1965).
- [109] STRASSEN, V.: Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten, *Num. Math.* **20** (1973), 238–251.

- [110] STRASSEN, V.: The computational complexity of continued fractions, *SIAM J. Comp.* **12** (1983), 1–27.
- [111] TARSKI, A.: *A decision method for elementary algebra and geometry*, University of California Press (1951).
- [112] THOM, R.: *Sur l'homologie des variétés algébriques réelles*, Differential and Combinatorial Topology (S.S. Cairns, ed.), Princeton Univ. Press (1965), 255–265.
- [113] TODA, S.: PP is as hard as the polynomial-time hierarchy, *SIAM J. Comp.* **21** (2) (1991), 865–877.
- [114] VALIANT, L.G.: The complexity of computing the permanent, *Theoret. Comp. Sci.* **8** (1979), 189–201.
- [115] VALIANT, L.G.: The complexity of enumeration and reliability problems, *SIAM J. Comp.* **8** (1979), 410–421.
- [116] WELSH, D.J.A.: *Complexity: Knots, Colourings and Counting*, Cambridge University Press (1994).
- [117] YAO, A.C.: On parallel computation for the knapsack problem, *J. ACM* **29** (1982), 898–903.
- [118] YAO, A.C.: Algebraic decision trees and Euler characteristic, *Proc. 33rd FOCS* (1992).
- [119] YAO, A.C.: Decision tree complexity and Betti numbers, *J. Comp. Syst. Sci.* **55** (1997), 36–43.