Counting Complexity Classes over the Reals I: The Additive Case*

Peter Bürgisser[1**] and Felipe Cucker[2***]

[1] Faculty of Computer Science, Electrical Engineering, and Mathematics
Paderborn University, D-33095 Paderborn, Germany
pbuerg@upb.de

[2] Department of Mathematics, City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong, China
macucker@math.cityu.edu.hk

Abstract. We define a counting class $\#\mathrm{P}_{\mathrm{add}}$ in the Blum-Shub-Smale-setting of additive computations over the reals. Structural properties of this class are studied, including a characterization in terms of the classical counting class #P introduced by Valiant. We also establish transfer theorems for both directions between the real additive and the discrete setting. Then we characterize in terms of completeness results the complexity of computing basic topological invariants of semi-linear sets given by additive circuits. It turns out that the computation of the Euler characteristic is $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$-complete, while for fixed k, the computation of the kth Betti number is $\mathrm{FPAR}_{\mathrm{add}}$-complete. Thus the latter is more difficult under standard complexity theoretic assumptions. We use all the above to prove some analogous completeness results in the classical setting.

1 Introduction

In 1989 Blum, Shub and Smale [4] introduced a theory of computation over the real numbers with the goal of providing numerical computations (as performed e.g., in numerical analysis or computational geometry) the kind of foundations classical complexity theory has provided to discrete computation. This theory describes the difficulty of solving numerical problems and provides a taxonomy of complexity classes capturing different degrees of such a difficulty.

Since its introduction, this BSS-theory has focused mainly on decisional problems. Functional problems attracted attention at the level of analysis of particular algorithms, but structural properties of classes of such problems were hardly studied. So far, the only systematic approach to study the complexity of certain functional problems within a framework of computations over the reals is Valiant's theory of VNP-completeness [6, 30, 33]. However, the relationship of this theory to the more general BSS-setting is, as of today, poorly understood.

* To appear in Proc. ISAAC 2003, Dec. 15 - 17, Kyoto, ©Springer
** Research supported by DFG grant BU 1371
*** Research supported by SRG grant 7001290

A detailed account of the research on complexity of real functions within the classical framework can be found in [14].

A first step in the study of functional properties could focus on complexity classes related to counting problems, i.e., functional problems, whose associated functions count the number of solutions of some decisional problem.

In classical complexity theory, counting classes were introduced by Valiant in his seminal papers [31, 32]. Valiant defined $\#\mathsf{P}$ as the class of functions which count the number of accepting paths of nondeterministic polynomial time machines and proved that the computation of the permanent is $\#\mathsf{P}$-complete. This exhibited an unexpected difficulty for the computation of a function, whose definition is only slightly different from that of the determinant, a problem known to be in $\mathsf{FNC}^2 \subseteq \mathsf{FP}$, and thus considered "easy." This difficulty was highlighted by a result of Toda [28] proving that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$, i.e., that $\#\mathsf{P}$ has at least the power of the polynomial hierarchy.

In the continuous setting, i.e., over the reals, the only attempt to define counting classes was made by Meer [17]. He defined a real version of the class $\#\mathsf{P}$ and studied some of its logical properties in terms of metafinite model theory. Meer did not investigate complete problems for this class.

In this paper we will define and study counting classes in the model of additive BSS-machines [15]. The computation nodes of these machines perform additions and subtractions, but no multiplications and divisions. The corresponding complexity classes are denoted by $\mathrm{P}_{\mathrm{add}}$ and $\mathrm{NP}_{\mathrm{add}}$[3].

The results in this paper can be seen as a first step towards a better understanding of the power of counting in the unrestricted BSS-model over the reals (allowing also for multiplications and divisions). A sequel to this paper studying this setting is under preparation [8].

Our results can be grouped in two kinds: structural relationships between complexity classes and completeness results. The latter (for whose proofs the former are used) satisfy a driving motivation for this paper: to capture the complexity of computing basic topological invariants of geometric objects in terms of complexity classes and completeness results.

Due to space limitations, proofs had to be omitted in this extended abstract.

2 Counting Classes and Completeness

Recall that $\#\mathsf{P}$ is the class of functions $f\colon \{0,1\}^\infty \to \mathbb{N}$ for which there exists a polynomial time Turing machine M and a polynomial p with the property that for all $n \in \mathbb{N}$ and all $x \in \{0,1\}^n$, $f(x)$ counts the number of strings $y \in \{0,1\}^{p(n)}$ such that M accepts (x, y).

By replacing Turing machines with additive BSS-machines in the above definition, we get a class of functions $f : \mathbb{R}^\infty \to \mathbb{N} \cup \{\infty\}$, which we denote by $\#\mathrm{P}_{\mathrm{add}}$. Thus $f(x)$ counts the number of vectors $y \in \mathbb{R}^{p(n)}$ such that M accepts

[3] To distinguish between classical and additive complexity complexity classes, we use the subscript "add" to indicate the latter. Also, to further emphasize this distinction, we write the former in sans serif.

(x, y). By counting only the number of "digital" vectors $y \in \{0,1\}^{p(n)}$, we obtain a smaller class of functions $f : \mathbb{R}^\infty \to \mathbb{N}$ denoted by $\mathrm{D\#P_{add}}$.

Completeness will be studied either with respect to parsimonious reductions or with respect to Turing reductions (see [24] for definitions).

We can prove that there is a wealth of natural complete problems for the class $\mathrm{D\#P_{add}}$ with respect to Turing reductions. This follows from the following general principle:

Proposition 1. *Let* $f : \mathbb{R}^\infty \to \mathbb{N}$ *belong to* $\mathrm{D\#P_{add}}$ *and assume that the restriction of* f *to* $\mathbb{Z}^\infty$ *is* $\#\mathsf{P}$*-complete with respect to Turing reductions. Then* f *is* $\mathrm{\#P_{add}}$*-complete and thus* $\mathrm{D\#P_{add}}$*-complete with respect to Turing reductions.*

The proof is based on the results of Sect. 3 and extends previous results due to Fournier and Koiran [11].

Proposition 1 yields plenty of Turing complete problems in $\mathrm{D\#P_{add}}$. We just mention two particularly interesting ones. Assume that we are given a graph G with real weights on the edges and some $w \in \mathbb{R}$. We define the weight of a subgraph as the sum of the weights of its edges.

1. (*Counting Traveling Salesman*) Let $\#\mathrm{TSP}_\mathbb{R}$ be the problem to count the number of Hamilton cycles of weight at most w in the graph G.
2. (*Counting Weighted Perfect Matchings*) Let $\#\mathrm{PM}_\mathbb{R}$ be the problem to count the number of perfect matchings of weight at most w in the graph G (here we assume that G is bipartite).

Valiant [31, 32] proved the $\#\mathsf{P}$-completeness of the problem to count the number of Hamilton cycles of a given graph, and of the problem to count the number of perfect matchings of a given bipartite graph. Together with Proposition 1 this immediately implies the following.

Corollary 1. *The problems* $\#\mathrm{TSP}_\mathbb{R}$ *and* $\#\mathrm{PM}_\mathbb{R}$ *are* $\mathrm{D\#P_{add}}$*-complete with respect to Turing reductions.*

3 The Power of Discrete Oracles and Boolean Parts

One can assign to any complexity class $\mathcal{C}$ of decision problems a corresponding counting complexity class $\# \cdot \mathcal{C}$ as follows [29]. To a set $A \in \{0,1\}^\infty$ and a polynomial p we assign the function $\#_A^p : \{0,1\}^\infty \to \mathbb{N}$, which associates to $x \in \{0,1\}^n$ the number $\#_A^p(x) := |\{y \in \{0,1\}^{p(n)} \mid (x,y) \in A\}|$. To a complexity class $\mathcal{C}$ of decision problems we assign $\# \cdot \mathcal{C} := \{\#_A^p \mid A \in \mathcal{C} \text{ and } p \text{ a polynomial}\}$. Similarly, one assigns $\# \cdot \mathcal{C}$ and $\mathrm{D}\# \cdot \mathcal{C}$ to a complexity class $\mathcal{C}$ over $\mathbb{R}$. Note that $\# \cdot \mathsf{P} = \#\mathsf{P}$, $\# \cdot \mathrm{P_{add}} = \mathrm{\#P_{add}}$, and $\mathrm{D}\# \cdot \mathrm{P_{add}} = \mathrm{D\#P_{add}}$.

The following result says that for several classical[4] complexity classes $\mathcal{C}$ consisting of decisional problems, the corresponding additive complexity class $\mathcal{C}_{\mathrm{add}}$

[4] All along this paper we use the words *discrete*, *classical* or *Boolean* to emphasize we are refering to the theory of complexity over a finite alphabet as exposed in, e.g., [1, 24].

is contained in, or even equal to, $\mathrm{P}_{\mathrm{add}}^{\mathcal{C}}$. That is, all problems in $\mathcal{C}_{\mathrm{add}}$ can be solved by an additive machine working in polynomial time and having access to a (discrete) oracle in $\mathcal{C}$. Likewise, if $\mathcal{C}$ is a classical complexity class of functions $\{0,1\}^\infty \to \{0,1\}^\infty$, we obtain that $\mathcal{C}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\mathcal{C}}$.

Theorem 1. *The following statements hold* ($k \geq 0$):

1. $\Sigma_{\mathrm{add}}^k \subseteq \mathrm{P}_{\mathrm{add}}^{\Sigma^k}$, $\Pi^{\mathrm{k}}{}_{\mathrm{add}} \subseteq \mathrm{P}_{\mathrm{add}}^{\Pi^k}$, $\mathrm{PH}_{\mathrm{add}} = \mathrm{P}_{\mathrm{add}}^{\mathsf{PH}}$.
2. $\mathrm{D\#} \cdot \Sigma_{\mathrm{add}}^k \subseteq \mathrm{FP}_{\mathrm{add}}^{\# \cdot \Sigma^k}$, $\mathrm{D\#} \cdot \Pi_{\mathrm{add}}^k \subseteq \mathrm{FP}_{\mathrm{add}}^{\# \cdot \Pi^k}$, $\mathrm{D\#} \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\# \cdot \mathsf{PH}}$.
3. $\mathrm{PAR}_{\mathrm{add}} = \mathrm{P}_{\mathrm{add}}^{\mathsf{PSPACE}}$.
4. $\mathrm{FPAR}_{\mathrm{add}} = \mathrm{FP}_{\mathrm{add}}^{\mathrm{PAR}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\mathsf{PSPACE}}$.

This result was already stated and proved for the class $\mathrm{NP}_{\mathrm{add}}$ in Fournier and Koiran [11]. Moreover, in [11, Remark 2] it was mentioned that the result for $\mathrm{NP}_{\mathrm{add}}$ can be extended to the classes of the polynomial hierarchy and to $\mathrm{PAR}_{\mathrm{add}}$. So what is new in Theorem 1 is the extension to the counting classes, and to the functional class $\mathrm{FPAR}_{\mathrm{add}}$. As in [10, 11], the proof relies on Meyer auf der Heide's (nonuniform) construction of small depth linear decision trees for point location in arrangements of hyperplanes [19, 20], see also Meiser [18].

An interesting application of the above insights is that Toda's famous result [28], as well as its extension by Toda and Watanabe [29], carries over to the real additive setting.

Corollary 2. *We have* $\mathrm{D\#} \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$.

We use this to prove that the counting class $\#\mathrm{P}_{\mathrm{add}}$ is closely related to its digital variant $\mathrm{D\#P}_{\mathrm{add}}$, in the sense that a $\#\mathrm{P}_{\mathrm{add}}$-oracle does not give more power to an additive polynomial time Turing machine than a $\mathrm{D\#P}_{\mathrm{add}}$-oracle

Theorem 2. *We have* $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\mathrm{D\#P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$. *In particular, a counting problem in* $\mathrm{D\#P}_{\mathrm{add}}$ *is* $\mathrm{D\#P}_{\mathrm{add}}$*-complete with respect to Turing reductions iff it is* $\#\mathrm{P}_{\mathrm{add}}$*-complete with respect to Turing reductions.*

The Boolean part of $\mathrm{D\#P}_{\mathrm{add}}$ consists of the restrictions of all functions in $\mathrm{D\#P}_{\mathrm{add}}$ to the set of binary inputs $\{0,1\}^\infty$. The proof of the following proposition is similar to that of [3, Theorem 2, Chap. 22].

Proposition 2. *We have* $\mathrm{BP}(\mathrm{D\#P}_{\mathrm{add}}) = \#\mathsf{P}/\mathsf{poly}$.

An application of our structural insights is the following transfer result.

Corollary 3. *We have the following transfer results:*

1. $\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}$ *iff* $\mathrm{D\#P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}$ *iff* $\#\mathsf{P} \subseteq \mathsf{FP}/\mathsf{poly}$.
2. $\mathrm{FPAR}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$ *iff* $\mathsf{PSPACE} \subseteq \mathsf{P}^{\#\mathsf{P}}/\mathsf{poly}$.
3. *Similar equivalences hold for additive machines without constants and uniform classical complexity classes, respectively.*

4 Complexity to Compute Topological Invariants

Algebraic topology studies topological spaces X by assigning to X various algebraic objects in a functorial way. In particular, homeomorphic (or even homotopy equivalent) spaces lead to isomorphic algebraic objects. For a general reference in algebraic topology we refer to [13, 23].

Typical examples of such algebraic objects studied are the (singular) homology vector spaces $H_k(X;\mathbb{Q})$ over $\mathbb{Q}$, defined for integers $k \in \mathbb{N}$. The dimension $b_k(X)$ of $H_k(X;\mathbb{Q})$ is called the kth *Betti number* of the space X. The zeroth Betti number $b_0(X)$ counts the number of connected components of X, and for $k > 0$, $b_k(X)$ measures a more sophisticated "degree of connectivity". Intuitively speaking, for a three-dimensional space X, $b_1(X)$ counts the number of holes and $b_2(X)$ counts the number of cavities of X. It is known that $b_k(X) = 0$ for $k > n := \dim X$. The Betti numbers modulo a prime p are defined by replacing the coefficient field $\mathbb{Q}$ by the finite field $\mathbb{F}_p$.

The *Euler characteristic* of X defined by $\chi(X) := \sum_{k=0}^{n}(-1)^k b_k(X)$ is an important numerical invariant of X, enjoying several nice properties. For a finite set X, $\chi(X)$ is just the cardinality of X.

The notion of a cell complex [13, 23] will be of importance for our algorithm to compute the Euler characteristic and the Betti numbers. For instance, if X is decomposed as a finite cell complex having c_k cells of dimension k, then $\chi(X) := \sum_{k=0}^{n}(-1)^k c_k(X)$.

We remark that the number of connected components, the Euler characteristic, and the Betti numbers lead to interesting lower complexity bounds for semi-algebraic decision problems, see [2, 34, 35] and the survey [7].

4.1 Semi-linear Sets and Additive Circuits

In this paper, we will confine our investigations to *semi-linear sets* $X \subseteq \mathbb{R}^n$, which are derived from closed halfspaces by taking a finite number of unions, intersections and complements. Moreover, we assume that the closed halfspaces are given by linear inequalities of "mixed type" $a_1X_1 + \cdots + a_nX_n \leq b$ with integer coefficients a_i and real right-hand side b.

We will represent semi-linear sets by a very compact data structure. An *additive circuit* $\mathcal{C}$ is a special arithmetic circuit [12], whose set of arithmetic operations is restricted to additions and subtractions. The circuit may have selection gates and use a finite set of real constants. The set of inputs accepted by an additive circuit is semi-linear, and any semi-linear set can be described this way. (See [3] for details.)

The basic problem $\mathrm{CSAT}_{\mathrm{add}}$ to decide whether a given semi-linear set X is nonempty turns out to be $\mathsf{NP}_{\mathrm{add}}$-complete [3]. By contrast, the feasibility question for a system of linear inequalities of the above mixed type is solvable in $\mathrm{P}_{\mathrm{add}}$. This is just a rephrasing of a well-known result by Tardos [27].

Over the real numbers, space is not as meaningful a resource as it is in the discrete setting [21]. The role of space, however, is satisfactorily played by parallel time formalized by the notion of uniform arithmetic circuits [4, 9]. We denote by $\mathrm{PAR}_{\mathrm{add}}$ the class of decision problems for which there exists a $\mathrm{P}_{\mathrm{add}}$-uniform family $(\mathcal{C}_n)$ of additive circuits such that the depth of $\mathcal{C}_n$ grows at most polynomially in n. $\mathrm{FPAR}_{\mathrm{add}}$ denotes the class of functions f which can be computed with such resources and such that the size of $f(x)$ is polynomially bounded in the size of x. (The size of a vector is defined as its length.)

In the computational problems listed below, it is always assumed that the input is an additive circuit $\mathcal{C}$ and X is the semi-linear set accepted by $\mathcal{C}$. We also say that X is defined or given by $\mathcal{C}$.

4.2 Complexity to Compute the Dimension

Our first results deals with the computation of the dimension. For all $d \geq 0$, the problem $\mathrm{DIM}_{\mathrm{add}}(d)$ consists of deciding whether the set X given by an additive circuit $\mathcal{C}$ has dimension at least d. We define $\dim \emptyset := -1$ so that we can decide for nonemptyness using the dimension function.

Theorem 3. *For all $d \geq 0$, the problem $\mathrm{DIM}_{\mathrm{add}}(d)$ is $\mathrm{NP}_{\mathrm{add}}$-complete.*

4.3 Counting Connected Components

Consider the *reachability problem* $\mathrm{REACH}_{\mathrm{add}}$ to decide for a given additive circuit $\mathcal{C}$ and two points s and t, whether these points are in the same connected component of the semi-linear set X defined by $\mathcal{C}$. The corresponding counting problem $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$ is the problem of counting the number of connected components of X given by $\mathcal{C}$.

Theorem 4. *The problems $\mathrm{REACH}_{\mathrm{add}}$ and $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$ are $\mathrm{PAR}_{\mathrm{add}}$-complete and $\mathrm{FPAR}_{\mathrm{add}}$-complete with respect to Turing reductions, respectively.*

The lower bound in this result is inspired by an early paper by Reif [25] (see also [26]), which showed the PSPACE-hardness of a generalized movers problem in robotics. Reif's result implies that the analogue of $\mathrm{REACH}_{\mathrm{add}}$ for semi-algebraic sets given by inequalities of (nonlinear) rational polynomials is PSPACE-hard. We cannot apply this result in our context, since we are dealing here with linear polynomials (of mixed type). However, we borrow from Reif's proof the idea to describe PSPACE by symmetric polynomial space Turing machines [16]. Roughly speaking, this is a nondeterministic Turing machine with the property that its transition relation is symmetric. Thus its configuration digraph is in fact a graph, which is essential for capturing the symmetric reachability relation of $\mathrm{REACH}_{\mathrm{add}}$.

4.4 Euler Characteristic and Betti Numbers

Let $\mathrm{EULER}_{\mathrm{add}}$ denote the following problem: given an additive circuit $\mathcal{C}$ defining a closed semi-linear set X, compute the Euler characteristic $\chi(X)$ of X. Hence only circuits defining closed semi-linear sets are considered to be admissible inputs. By convention, we assume that $\chi(\emptyset) := \infty$, so that we can distinguish empty sets from nonempty ones.

The next lemma tells us that both closedness and compactness of a semi-linear set can be checked within the allowed resources.

Lemma 1. *Both closedness and compactness of a set X given by an additive circuit can be decided in* $\mathrm{P}_{\mathrm{add}}^{\#\mathsf{P}}$.

Theorem 5. *The problem* $\mathrm{EULER}_{\mathrm{add}}$ *is* $\mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$*-complete with respect to Turing reductions.*

For $k \in \mathbb{N}$, we define $\mathrm{BETTI}_{\mathrm{add}}(k)$ to be the problem of computing the kth Betti number of a closed semi-linear set given by an additive circuit. The problem of computing the kth Betti number modulo a prime p shall be denoted by $\mathrm{BETTI}_{\mathrm{add}}(k, \bmod p)$. Note that for $k = 0$, these are just the problems of counting the number of connected components, respectively counting them modulo p.

The following result extends Theorem 4.

Theorem 6. *For any $k \in \mathbb{N}$ and any prime p, the problems* $\mathrm{BETTI}_{\mathrm{add}}(k)$ *and* $\mathrm{BETTI}_{\mathrm{add}}(k, \bmod p)$ *are* $\mathrm{FPAR}_{\mathrm{add}}$*-complete with respect to Turing reductions.*

The $\mathrm{PAR}_{\mathrm{add}}$-hardness of $\mathrm{BETTI}_{\mathrm{add}}(k)$ is deduced from Theorem 4 by establishing a parsimonious reduction from $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$ to $\mathrm{BETTI}_{\mathrm{add}}(k)$. This can be done using the suspension construction of algebraic topology. A short outline of the upper bound proof is given in Sect. 4.6.

These results give a complexity theoretic distinction between the problems to compute the Euler characteristic and to compute Betti numbers. The computation of the Euler characteristic is strictly easier than the computation of the number of connected components, or more generally than the computation of the kth Betti number for any fixed k, unless the unlikely collapse $\mathsf{FPSPACE} = \mathsf{FP}^{\#\mathsf{P}}$ happens. Intuitively, the fact that $\mathrm{EULER}_{\mathrm{add}}$ is easier than $\mathrm{BETTI}_{\mathrm{add}}(k)$ can be explained by the various nice properties satisfied by the Euler characteristic.

4.5 Completeness Results in the Turing Model

Let us now restrict the inputs in the three problems $\mathcal{P}$ studied above to *constant free* additive circuits and denote the resulting computational problem by $\mathcal{P}^0$. Note that constant-free circuits can be encoded over a finite alphabet and thus be handled by (classical) Turing machines.

Corollary 4. *For any $k, d \in \mathbb{N}$ and any prime p,* $\mathrm{DIM}_{\mathrm{add}}^0(d)$ *is* NP*-complete,* $\mathrm{EULER}_{\mathrm{add}}^0$ *is* $\mathsf{FP}^{\#\mathsf{P}}$*-complete, and* $\mathrm{BETTI}_{\mathrm{add}}^0(k)$*,* $\mathrm{BETTI}_{\mathrm{add}}^0(k, \bmod p)$ *both are* $\mathsf{FPSPACE}$*-complete.*

4.6 Computing Betti Numbers mod 2 in FPSPACE

We outline the proof of $\mathrm{BETTI}^0_{\mathrm{add}}(k, \mathrm{mod}\, 2) \in \mathsf{FPSPACE}$. Working modulo 2 has the advantage that we do not have to worry about orientations.

For $s, n \in \mathbb{N}$ we define $\mathcal{H}_{s,n}$ to be the set of affine linear polynomials $a_0 + \sum_{i=1}^n a_i X_i$ with integer coefficients a_i such that $\sum_{i=0}^n |a_i| \le 2^s$. We denote by $\mathcal{F}_{s,n}$ the set of all non-empty sets $F = \bigcap_{f \in \mathcal{H}_{s,n}} \{x \in \mathbb{R}^n \mid f(x) = \sigma(f)\}$, corresponding to some sign function $\sigma\colon \mathcal{H}_{s,n} \to \{-1, 0, 1\}$. The space $\mathbb{R}^n$ is the disjoint union of all $F \in \mathcal{F}_{s,n}$. We will call this the *universal cell decomposition* for the parameters s, n, and we call the sets $F \in \mathcal{F}_{s,n}$ the corresponding *faces* or *cells*.

It is known [4, Thm. 3, Chapt. 21] that each face $F \in \mathcal{F}_{s,n}$ contains a rational point of bit-size at most $(sn)^c$, for some fixed constant $c > 0$. We shall therefore encode a face $F \in \mathcal{F}_{s,n}$ by a triple $(s, n, x) \in \mathbb{N}^2 \times \mathbb{Q}$ such that $x \in F$ and the bit-size of x is at most $(sn)^c$. (Note that one may need exponentially many inequalities to describe a face F.)

If $X \subseteq \mathbb{R}^n$ is compact and a finite union of faces in $\mathcal{F}_{s,n}$, then the decomposition of X is a finite cell complex. The important thing to check here is that the boundary of a cell in $\mathcal{F}_{s,n}$ is a union of cells in $\mathcal{F}_{s,n}$.

Let $\mathcal{C}$ be an additive circuit of size s that defines a compact semi-linear set $X \subseteq \mathbb{R}^n$. We assume that $\mathcal{C}$ uses only the constants $0, 1$ and that it branches according to the sign of intermediate results in a ternary way (< 0, $= 0$, > 0). The set of inputs in $\mathbb{R}^n$, whose path in the corresponding computation tree ends up with a specific leaf ν, shall be called the *leaf set* D_ν of ν. Note that each nonempty leaf set D_ν of $\mathcal{C}$ is union of faces of $\mathcal{F}_{s,n}$, where s denotes the size of $\mathcal{C}$. Therefore, $\{F \in \mathcal{F}_{s,n} \mid F \subseteq X\}$ is a finite cell complex.

We are going to define the cellular homology of this cell complex with respect to the coefficient field $\mathbb{F}_2$. Two faces F, F' of $\mathcal{F}_{s,n}$ are called *incident* iff F' is contained in the closure of F, and if the dimensions of F and F' differ exactly bye one. Let Φ_k denote the set of the k-cells of the cell complex and $\mathcal{C}_k$ be the $\mathbb{F}_2$-vector space having Φ_k as a basis. The boundary map $\partial_k\colon \mathcal{C}_{k+1} \to \mathcal{C}_k$ is the $\mathbb{F}_2$-linear map defined for $F \in \Phi_{k+1}$ by $\partial_k(F) = \sum_{F'} [F, F']\, F'$, where the sum is over all $F' \in \Phi_k$ incident to F. The image $B_k := \mathrm{im}\partial_k$ of ∂_k is called the vector space of k-boundaries, and the kernel $Z_k := \ker \partial_{k-1}$ is called the space of k-cycles. The kth *cellular homology vector space* is defined as the quotient space $H_k := Z_k/B_k$. It is well known [13, 23] that H_k is isomorphic to the singular homology vector space $H_k(X; \mathbb{F}_2)$. Therefore, $b_k := \dim H_k$ is the kth Betti number modulo 2, which is independent of the cell decomposition. We have $b_k = \dim Z_k - \dim B_k = c_k - \rho_{k-1} - \rho_k$, where $\rho_k := \mathrm{rank}\, \partial_k$ and $c_k = |\Phi_k|$.

Lemma 2. *The incidence of faces F, F' in $\mathcal{F}_{s,n}$, $s, n \in \mathbb{N}$, can be decided in* PH.

We make now a short digression on space efficient linear algebra. By a *succinct representation* of an integer matrix $A = (a_{ij})$ we understand a Boolean circuit computing the matrix entry a_{ij} from the index pair (i, j) given in binary.

Combing the results of [22] and [5], we obtain:

Lemma 3. *The rank of an $N \times N$ integer matrix A given in succinct representation by a Boolean circuit B can be computed by a Turing machine with space polynomial in $\log N$, the depth of B, and the log of the maximal bitsize of the elements of A.*

The proof that $\mathrm{BETTI}^0_{\mathrm{add}}(k, \mathrm{mod} p)$ is contained in $\mathsf{FPSPACE}$ follows by combining Lemma 2 with Lemma 3.

5 Open questions

Problem 1. In this paper, we prove completeness with respect to Turing reductions. Do we also have completeness with respect to parsimonious reductions? For instance, how about the completeness of $\#\mathrm{TSP}_{\mathbb{R}}$ in $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$?

Problem 2. What is the complexity to deciding connectedness of a semi-linear set given by an additive circuit?

Problem 3. In this paper we proved that computing the torsion-free part of the homology of semi-linear sets is $\mathrm{FPAR}_{\mathrm{add}}$-complete. What is the complexity of computing the torsion part of this homology?

Acknowledgments. We thank Eric Allender and Saugata Basu for helpful discussions. Also, we are grateful to anonymous referees for many valuable comments, which helped to improve the presentation significantly.

References

1. J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* EATCS Monographs on Theoretical Computer Science, 11. Springer-Verlag, 1988.
2. M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th ACM STOC, Boston*, pages 80–86, 1983.
3. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer, 1998.
4. L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.
5. A.B. Borodin. On relating time and space to size and depth. *SIAM J. Comp.*, 6:733–744, 1977.
6. P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics.* Springer Verlag, 2000.
7. P. Bürgisser. Lower bounds and real algebraic geometry. In S. Basu and L. Gonzales-Vega, editors, *Algorithmic and Quantitative Real Algebraic Geometry*, volume 60 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science.* AMS, 2003.
8. P. Bürgisser and F. Cucker. Counting complexity classes over the reals. II: The unrestricted case. In preparation.
9. F. Cucker and P. Koiran. Computing over the reals with addition and order: Higher complexity classes. *J. Compl.*, 11:358–376, 1995.

10. H. Fournier and P. Koiran. Are lower bounds easier over the reals? In *Proc. 30th ACM STOC*, pages 507–513, 1998.
11. H. Fournier and P. Koiran. Lower bounds are not easier over the reals: Inside PH. In *Proc. ICALP 2000*, LNCS 1853, pages 832–843, 2000.
12. J. von zur Gathen. Parallel arithmetic computations: a survey. In *Proc. 12th Symp. Math. Found. Comput. Sci., Bratislava*, number 233 in LNCS, pages 93–112, 1986.
13. Allen Hatcher. *Algebraic topology.* Cambridge University Press, Cambridge, 2002.
14. K.-I. Ko. *Complexity of Real Functions.* Birkhäuser, 1991.
15. P. Koiran. Computing over the reals with addition and order. *Theoret. Comp. Sci.*, 133:35–47, 1994.
16. H.R. Lewis and C.H. Papadimitriou. Symmetric space-bounded computation. *Theoret. Comp. Sci.*, 19:161–187, 1982.
17. K. Meer. Counting problems over the reals. *Theoret. Comp. Sci.*, 242:41–58, 2000.
18. S. Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106:286–303, 1993.
19. F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *J. ACM*, 31:668–676, 1984.
20. F. Meyer auf der Heide. Fast algorithms for n-dimensional restrictions of hard problems. *J. ACM*, 35:740–747, 1988.
21. C. Michaux. Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum, Shub et Smale. *C. R. Acad. Sci. Paris*, 309, Série I:435–437, 1989.
22. K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7:101–104, 1987.
23. James R. Munkres. *Elements of algebraic topology.* Addison-Wesley Publishing Company, Menlo Park, CA, 1984.
24. C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
25. J.H. Reif. Complexity of the mover's problem and generalizations. In *Proc. 20th FOCS*, pages 421–427, 1979.
26. J.H. Reif. Complexity of the generalized mover's problem. In J.T. Schwartz, M. Sharir, and J. Hopcroft, editors, *Planning, Geometry and Complexity of Robot Motion*, pages 267–281. Ablex Publishing Corporation, 1987.
27. E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34:250–256, 1986.
28. S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comp.*, 21(2):865–877, 1991.
29. S. Toda and O. Watanabe. Polynomial time 1-Turing reductions from #ph to #p. *Theoret. Comp. Sci.*, 100:205–221, 1992.
30. L.G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM STOC*, pages 249–261, 1979.
31. L.G. Valiant. The complexity of computing the permanent. *Theoret. Comp. Sci.*, 8:189–201, 1979.
32. L.G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comp.*, 8:410–421, 1979.
33. L.G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic: an International Symposium held in honor of Ernst Specker*, volume 30, pages 365–380. Monogr. No. 30 de l'Enseign. Math., 1982.
34. A.C. Yao. Algebraic decision trees and Euler characteristic. In *Proc. 33rd FOCS*, 1992.
35. A.C. Yao. Decision tree complexity and Betti numbers. In *Proc. 26th ACM STOC*, 1994.