COUNTING COMPLEXITY CLASSES FOR NUMERIC COMPUTATIONS I: SEMI-LINEAR SETS*

PETER BÜRGISSER† AND FELIPE CUCKER‡

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 $\#\mathsf{P}$ 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.

Key words. counting complexity, real complexity classes, Euler characteristic, Betti numbers

AMS subject classifications. 68Q15,68Q17, 55-04

1. Introduction. In 1989 Blum, Shub and Smale [5] 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 [7, 40, 43]. However, the relationship of this theory to the more general BSS-setting is, as of today, poorly understood. A detailed account of the research on complexity of real functions within the classical framework can be found in [23].

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 [41, 42]. 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 [38] 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

*This work has been supported by SRG grant 7001558 and DFG grant BU 1371.

†Faculty of Computer Science, Electrical Engineering, and Mathematics, Paderborn University, D-33095 Paderborn, Germany (pbuerg@upb.de).

‡Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong (macucker@math.cityu.edu.hk).

classes was made by Meer [27]. He defined a real version of the class #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 [24]. 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 $\mathrm{NP}_{\mathrm{add}}$[1].

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 [11].

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. In the following, we give an outline of the main results of this paper.

1.1. Counting classes. Recall that #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 (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 $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$.

In Theorem 4.7 we show that a counting problem $f \in \mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ is $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-complete with respect to Turing reductions iff it is $\#\mathrm{P}_{\mathrm{add}}$-complete with respect to Turing reductions. Moreover, in §4.3 we prove that there is a wealth of natural complete problems for the class $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ with respect to Turing reductions. For instance, consider the following counting version of the real weighted perfect matching problem $\#\mathrm{PM}_{\mathbb{R}}$: given $w \in \mathbb{R}$ and a bipartite graph G with real weights on the edges, count the number of perfect matchings of G having weight at most w. (The weight of a matching is defined as the sum of the weights of its edges.) The problem $\#\mathrm{PM}_{\mathbb{R}}$ turns out to be $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-complete with respect to Turing reductions. The same is true for the counting version of the real traveling salesman problem $\#\mathrm{TSP}_{\mathbb{R}}$ to count the number of Hamilton circuits of weight at most w of a given graph with real weights on the edges. It is an interesting open problem whether these problems are also $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-complete with respect to parsimonious reductions, see §7.

The above completeness results follow from a general principle (Proposition 4.13), which says that for proving $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-completeness of the counting version of a problem in $\mathrm{DNP}_{\mathrm{add}}$, it is sufficent to show that the restriction of the corresponding counting problem to integer inputs is #P-complete. The proof of this principle is based on an extension of the structural relationships between the real additive and the discrete setting, discovered by Fournier and Koiran [16, 17], discussed next.

[1] 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.

1.2. Structural relationships. The main result of §4.1 is summarized in Theorem 4.1, which says that for several classical[2] complexity classes $\mathcal{C}$ consisting of decisional problems, the corresponding additive complexity class $\mathcal{C}_{\mathrm{add}}$ is contained in, or even equal to, $\mathrm{P}^{\mathcal{C}}_{\mathrm{add}}$. 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 $\mathcal{C}_{\mathrm{add}} \subseteq \mathrm{FP}^{\mathcal{C}}_{\mathrm{add}}$. In particular, we have $\mathrm{FP}^{\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}} = \mathrm{FP}^{\#\mathsf{P}}_{\mathrm{add}}$.

Theorem 4.1 is an extension of the work of Fournier and Koiran [16, 17], who discovered this close relationship between the real additive and the discrete setting. This relationship is based on Meyer auf der Heide's (nonuniform) construction of small depth linear decision trees for point location in arrangements of hyperplanes [29, 30] (see also Meiser [28] for an extension of these results). Fournier and Koiran showed that this construction can be made uniform if a classical NP-oracle is available. This way, they proved that $\mathrm{NP}_{\mathrm{add}} \subseteq \mathrm{P}^{\mathsf{NP}}_{\mathrm{add}}$. We have extended this result in various directions, in particular to the counting context.

An interesting application of the above insights is that Toda's famous result [38], as well as its extension by Toda and Watanabe's [39], carry over to the real additive setting (Corollary 4.6). We use this to prove that the counting class $\#\mathrm{P}_{\mathrm{add}}$ is closely related to its digital variant $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ in the sense that $\mathrm{FP}^{\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}} = \mathrm{FP}^{\mathrm{D}\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}}$. In other words, a $\#\mathrm{P}_{\mathrm{add}}$-oracle does not give more power to an additive polynomial time Turing machine than a $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-oracle.

An important application of our structural insights is the following transfer result (Corollary 4.11)

$$\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}} \iff \mathrm{D}\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}} \iff \#\mathsf{P} \subseteq \mathsf{FP}/\mathsf{poly}.$$

The proof uses the fact that the Boolean part of $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$, consisting of the restrictions of all functions in $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ to the set of binary inputs $\{0,1\}^\infty$, is equal to $\#\mathsf{P}/\mathsf{poly}$ (Proposition 4.10).

1.3. 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 [20, 33]. 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 *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 [20, 33] will be of importance for our algorithms 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 $\chi(X) := \sum_{k=0}^{n} (-1)^k c_k$.

[2] 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., [2, 34].

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 [3, 44, 45] and the survey [9].

1.4. 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 [19], 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. (See Definition 2.9 for details.) The set of inputs accepted by an additive circuit is semi-linear, and any semi-linear set can be described this way.

The basic problem $\mathrm{CSAT}_{\mathrm{add}}$ to decide whether the semi-linear set X given by an additive circuit is non-empty, turns out to be $\mathsf{NP}_{\mathrm{add}}$-complete [4]. 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 [37] (cf. Remark 2.7).

Over the real numbers, space is not as meaningful a resource as it is in the discrete setting (cf. [31]). The role of space, however, is satisfactorily played by parallel time formalized by the notion of uniform arithmetic circuits (cf. [4, 14]). 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 (see §2). $\mathrm{FPAR}_{\mathrm{add}}$ 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.)

1.5. Completeness results for topological invariants. 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 $\mathcal{C}$.

Among the completeness results proved in this paper, the most important ones are the following (for a complete list see §6).

1. The problem $\mathrm{DIM}_{\mathrm{add}}(d)$ to decide whether $\dim X \geq d$ is $\mathrm{NP}_{\mathrm{add}}$-complete (Theorem 5.1).

2. The problem $\mathrm{EULER}_{\mathrm{add}}$ to compute the Euler characteristic of a closed semi-linear set X is $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$-complete with respect to Turing reductions (Theorem 5.18).

3. The problem $\mathrm{BETTI}_{\mathrm{add}}(k)$ to compute the kth Betti number $b_k(X)$ of of a closed semi-linear set X is $\mathrm{FPAR}_{\mathrm{add}}$-complete with respect to Turing reductions (Theorem 5.19).

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, under a standard complexity theoretic assumption (Corollary 5.23). 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.

Let us now restrict the inputs in the above three problems $\mathcal{P}$ 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. In §5.3.6 we derive the following completeness results in the Turing model: $\mathrm{DIM}^0_{\mathrm{add}}(d)$ is NP-complete, $\mathrm{EULER}^0_{\mathrm{add}}$ is $\mathsf{FP}^{\#\mathsf{P}}$-complete, and $\mathrm{BETTI}^0_{\mathrm{add}}(k)$ is FPSPACE-complete.

We briefly describe the proof idea for $\mathrm{BETTI}_{\mathrm{add}}(k)$. The lower bound is inspired by an early paper by Reif [35] (see also [36]), who showed the PSPACE-hardness of a generalized movers problem in robotics. The *reachability problem* $\mathrm{REACH}_{\mathrm{add}}$ is the following: given an additive circuit defining the semi-linear set X and given two points $s, t \in X$, decide whether s and t are in the same connected component of X. Reif's result implies that the analogue of the reachability problem for semi-algebraic sets (given by inequalities of 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 characterize PSPACE by symmetric polynomial space Turing machines [26] and prove that $\mathrm{REACH}_{\mathrm{add}}$ is $\mathrm{PAR}_{\mathrm{add}}$-hard (Proposition 5.9). From this lower bound result, one can derive the $\mathrm{PAR}_{\mathrm{add}}$-hardness of $\mathrm{BETTI}_{\mathrm{add}}(k)$ by standard constructions of algebraic topology.

The proof that $\mathrm{BETTI}_{\mathrm{add}}(k)$belongs to $\mathrm{FPAR}_{\mathrm{add}}$ proceeds by the following steps: 1. An additive circuit $\mathcal{C}$ accepting a set X defines a decomposition of X into leaf sets. This decomposition can be refined to a finite cell complex if X is compact (cf. §5.3.3). 2. The matrices (a_{ij}) of the boundary maps of the corresponding cellular homology can be succinctly represented by Boolean circuits computing a_{ij} from the index pair (i, j) given in binary. 3. The rank of an integer matrix given in succinct representation can be computed in a space efficient manner (Lemma 5.21).

1.6. Organization of the paper. We start in §2 by introducing some notation and recalling basic facts about additive machines. Then we define in §3 the counting complexity class $\#\mathrm{P}_{\mathrm{add}}$ in the additive model as well as its digital variant $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$, introduce different notions of reductions, and prove some basic completeness results. Section 4 deals with structural relationships and can be seen as the first part of this paper. Section 5 about the complexity to compute topological invariants constitutes the second part of this paper. It contains completeness proofs for several natural computational problems, each of which are treated in separate subsections. Those problems are: counting connected components, computing the Euler characteristic, and computing Betti numbers. We also present completeness results for the corresponding problems in the Turing model in §5.3.6. Finally, we end the paper in §6 with a summary of problems and results, and with some selected open problems in §7.

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

2. Preliminaries about additive machines. We denote by $\mathbb{R}^\infty$ the disjoint union $\mathbb{R}^\infty = \bigsqcup_{n\geq 0}\mathbb{R}^n$, where for $n \geq 0$, $\mathbb{R}^n$ is the standard n-dimensional space over $\mathbb{R}$. The space $\mathbb{R}^\infty$ is a natural one to represent problem instances of arbitrarily high dimension. For $x \in \mathbb{R}^n \subset \mathbb{R}^\infty$, we call n the *size* of x and we denote it by $\mathrm{size}(x)$. Contained in $\mathbb{R}^\infty$ is the set of bitstrings $\{0,1\}^\infty$ defined as the union of the sets $\{0,1\}^n$, for $n \in \mathbb{N}$.

Additive machines (in the sequel called simply "machines") are BSS-machines whose computation nodes perform only additions and subtractions (see [4, 24] for details). To a machine M we naturally associate an input-output map $\varphi_M : \mathbb{R}^\infty \to \mathbb{R}^\infty$. We shall say that a function $f : \mathbb{R}^\infty \to \mathbb{R}^k$, $k \leq \infty$, is *computable* when there

is a machine M such that $f = \varphi_M$. Also, a set $A \subseteq \mathbb{R}^\infty$ is *decided* by a machine M if its characteristic function $\chi_A : \mathbb{R}^\infty \to \{0,1\}$ coincides with φ_M. So, for decision problems, we consider machines whose output space is $\{0,1\} \subset \mathbb{R}$.

We can now introduce some central complexity classes.

DEFINITION 2.1. *A machine M over $\mathbb{R}$ is said* to work in polynomial time *when there is a constant $c \in \mathbb{N}$ such that for every input $x \in \mathbb{R}^\infty$, M reaches its output node after at most* $\mathrm{size}(x)^c$ *steps. The class* $\mathrm{P}_{\mathrm{add}}$ *is then defined as the sets of all subsets of $\mathbb{R}^\infty$ that can be decided by a machine working in polynomial time. The class* $\mathrm{FP}_{\mathrm{add}}$ *is the class of functions computed by machines working in polynomial time.*

DEFINITION 2.2. *A set A belongs to* $\mathrm{NP}_{\mathrm{add}}$ *if there is a machine M satisfying the following condition: for all x, $x \in A$ iff there is $y \in \mathbb{R}^\infty$ such that M accepts the input (x, y) within time polynomial in* $\mathrm{size}(x)$*. In this case, the element y is said to be a* witness *for x. If we require the witness y to belong to $\{0,1\}^\infty$ we say that $A \in \mathrm{DNP}_{\mathrm{add}}$ (the D standing for* digital*). Abusing language we will call the machine M above an* $\mathrm{NP}_{\mathrm{add}}$-machine *(resp. a* $\mathrm{DNP}_{\mathrm{add}}$*-machine).*

REMARK 2.3.

(i) In this model the element y can be seen as the sequence of guesses used in the Turing machine model (but note that, in the case of $\mathrm{NP}_{\mathrm{add}}$, these guesses are not necessarily binary). However, we note that in this 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 w.l.o.g., the length of y can be bounded by the running time of M (which is of the form $p(\mathrm{size}(x))$ for a polynomial p).

(ii) The definitions of $\mathrm{NP}_{\mathrm{add}}$ and $\mathrm{DNP}_{\mathrm{add}}$ extend in a straightforward manner to all levels of the polynomial hierarchies $\mathrm{PH}_{\mathrm{add}}$ and $\mathrm{DPH}_{\mathrm{add}}$ respectively (i.e., to the classes Σ^k_{add} and Π^k_{add} for $k \geq 0$). For details see [4, 14].

DEFINITION 2.4. *We say that an additive machine* has no real constants *when the only machine constants appearing in its program are 0 and 1. Complexity classes for these machines are distinguished by the superscript 0 as in* $\mathrm{P}^0_{\mathrm{add}}$, $\mathrm{NP}^0_{\mathrm{add}}$.

Natural examples of sets in these classes exist. For instance, the real traveling salesman problem $\mathrm{TSP}_{\mathbb{R}}$ discussed in §1.1 belongs to $\mathrm{DNP}_{\mathrm{add}}$ (actually to $\mathrm{DNP}^0_{\mathrm{add}}$). Problems which are known to be $\mathrm{NP}_{\mathrm{add}}$-complete for many-one reductions are scarse (for some known problems see [15]). In contrast, the following result by Fournier and Koiran [17] exhibits plenty of $\mathrm{NP}_{\mathrm{add}}$-complete problems with respect to Turing reductions. For a problem $S \subseteq \mathbb{R}^\infty$, we define its *integer part* to be $S_{\mathbb{Z}} = S \cap \mathbb{Z}^\infty$.

THEOREM 2.5 ([17]). *Let $S \in \mathrm{NP}_{\mathrm{add}}$. If $S_{\mathbb{Z}}$ is* NP*-complete with respect to Turing reductions, then S is* $\mathrm{NP}_{\mathrm{add}}$*-complete with respect to Turing reductions.*

Of course, this theorem implies the $\mathrm{NP}_{\mathrm{add}}$-completeness for Turing reductions of a large number of decision problems, for instance for $\mathrm{TSP}_{\mathbb{R}}$ and $\mathrm{PM}_{\mathbb{R}}$ (for formal definitions of these problems, see §6). Note that, in particular, every discrete NP-complete problem (e.g., SAT) is $\mathrm{NP}_{\mathrm{add}}$-Turing-complete.

A basic fact used in proving many results on additive machines is the existence of "small" rational points in polyhedra when the defining matrix has "small" integer entries. In what follows, for an integer $n \geq 1$, we denote the set $\{1, \ldots, n\}$ by $[n]$.

THEOREM 2.6 (Theorem 3, Chapter 21 of [4]). *Let P be a non-empty polyhedron of $\mathbb{R}^n$ defined by a system*

$$A_1 y \leq b_1; \ A_2 y < b_2, \tag{2.1}$$

where $A_1 \in \mathbb{Z}^{N_1 \times n}, A_2 \in \mathbb{Z}^{N_2 \times n}, b_1 \in \mathbb{R}^{N_1}$, and $b_2 \in \mathbb{R}^{N_2}$. The entries of A_1 and A_2 are integers of bit-size bounded by L. Then there is $y \in P$ with the following

description

$$y_i = \sum_{j \in I_1} u_{ij} b_{1j} + \sum_{j \in I_2} v_{ij} b_{2j} + w_i,\ i = 1, \ldots, n$$

where $I_1 \subseteq [N_1]$, $I_2 \subseteq [N_2]$, $|I_1| + |I_2| \leq n$, *and the coefficients* u_{ij}, v_{ij}, w_i *are rationals of bit-size at most* $(Ln)^c$ *for some constant* c.

REMARK 2.7. The feasibility of a system (2.1) of linear inequalities for given integer matrices A_1, A_2 and real vectors b_1, b_2 can be decided in $\mathrm{P}_{\mathrm{add}}$. Moreover, a solution can be computed in $\mathrm{FP}_{\mathrm{add}}$, if it exists. This is just a rephrasing a of well-known and important result by Tardos [37]. We will not need this remark in the rest of the paper.

Recall from [4] or [10, Ex. 3.15] that a *linear decision tree* T is a regular binary tree, whose internal nodes are labeled by linear functions $\ell : \mathbb{R}^n \to \mathbb{R}$, and whose leaves are labeled with "accept" or "reject". Here, n is the dimension of the input space. At a given node, an input $x \in \mathbb{R}^n$ goes to the left child if $\ell(x) \geq 0$ and to the right child if $\ell(x) < 0$. Let $X \subseteq \mathbb{R}^n$ be the set accepted by the linear decision tree T. The set of inputs in $\mathbb{R}^n$, whose path in the computation tree T ends up with a specific leaf ν, shall be called the *leaf set* D_ν of ν. Note that the set D_ν can be described by a set of linear inequalities and is therefore convex. It is clear that the leaf sets corresponding to the accepting leaves form a partition of the set X. In particular, X is semi-linear (cf. §1.4).

We next use Theorem 2.6 to prove that $\mathrm{NP}_{\mathrm{add}} = \mathrm{DNP}_{\mathrm{add}}$. This is a well-known result [24], but the idea of the proof will be repeatedly used in this paper.

COROLLARY 2.8. $\mathrm{NP}_{\mathrm{add}} = \mathrm{DNP}_{\mathrm{add}}$.

Proof. Let $X \in \mathrm{NP}_{\mathrm{add}}$ and M be a machine deciding X as in Definition 2.2. By unwinding the computation of M on pairs $(x, y) \in \mathbb{R}^n \times \mathbb{R}^{p(n)}$ we obtain a linear decision tree T of depth polynomial in n. If z is a value tested for positivity at a branch node of this tree, then

$$z = \sum_{i=1}^{p(n)} a_i y_i + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{k} c_i \alpha_i + d, \tag{2.2}$$

where $\alpha_1, \ldots, \alpha_k$ are the constants of M and the coefficients a_i, b_i, c_i, and d are integers of bit-size polynomial in n. Thus, for a given $x \in \mathbb{R}^n$, the leaf set D_ν of points y such that (x, y) reaches the accepting leaf ν in T is the set of solutions of a system of inequalities as in Theorem 2.6. We conclude that D_ν is non-empty if and only if D_ν contains a point y such that, for $i = 1, \ldots, p(n)$,

$$y_i = \sum_{j=1}^{n} b_{ij} x_j + \sum_{j=1}^{k} c_{ij} \alpha_j + d_i \tag{2.3}$$

where the coefficients b_{ij}, c_{ij} and d_i are rationals of bit-size polynomial in n (for a polynomial which does not depend on ν or x). Then, to decide whether $x \in X$, one can guess $b_{ij}, c_{ij}, d_i \in \mathbb{Q}$, compute y_i according to (2.3) and check whether M accepts $(x_1, \ldots, x_n, y_1, \ldots, y_{p(n)})$. Alternatively, one could also compute y in $\mathrm{FP}_{\mathrm{add}}$ according to Remark 2.7 and check whether M accepts $(x_1, \ldots, x_n, y_1, \ldots, y_{p(n)})$. □

Over the real numbers, space is not as meaningful a resource as it is in the discrete setting (cf. [31]). The role of space, however, is satisfactorily played by parallel time

(cf. [4, 14]). To introduce parallel time, we briefly recall the model of additive circuits, a restriction of the more general model of arithmetic circuits introduced in [19].

DEFINITION 2.9. *An* additive circuit $\mathcal{C}$ *over* $\mathbb{R}$ *is an acyclic directed graph where each node has indegree 0, 1, 2 or 3. 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* $\{+, -\}$*. They are called* arithmetic nodes*. Nodes with indegree 1 are* output nodes*. Nodes with indegree 3 are* selection nodes*. All output nodes have outdegree 0. Otherwise, there is no upper bound on the outdegree of the other nodes.*

For an additive 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.*

The semantics of a selection node is as follows. With input (x, y, z) the node returns y if $x \geq 0$ and z otherwise. The semantics of all other nodes is obvious. If $\mathcal{C}$ is an additive circuit with n input nodes and m output nodes, we may talk about the function $\varphi_{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R}^m$ *computed by the circuit*. We remark that the computation of an additive circuit can always be unwound to a linear decision tree.

Let $f : \mathbb{R}^\infty \to \mathbb{R}^\infty$. The family of additive 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 $\mathbb{R}^n$.

The other ingredient we need to define parallel complexity classes, is a notion of uniformity.

Note that nodes of additive circuits can be described by five real numbers in the following way. If the nodes of the circuit are $g_1, \ldots, g_N$, then node g_j is described by the tuple $(j, t, i_\ell, i_r, i_m) \in \mathbb{R}^5$ where t represents the type of g_j according to the following (arbitrary) dictionary:

g_j	input	constant	+	−	selection	output
t	1	2	3	4	5	6

For nodes of indegree two, i_ℓ and i_r denote the numbers of the nodes which provide left and right input to g_j, respectively. If g_j is a constant node, then i_ℓ equals its constant and if g_j is an output node, then i_ℓ numbers the node which provides the input to g_j. Finally, if g_j is a selection node, then i_ℓ, i_r and i_m number its left, right and middle inputs. All components not mentioned above are set to 0. Thus, the whole circuit can be described by a point in $\mathbb{R}^{5N}$.

DEFINITION 2.10. *A family of circuits* $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ *is said to be* uniform *if there exists an additive machine* M *that, on input* (n, i)*, outputs the description of the* i*-th node of* $\mathcal{C}_n$*. If* M *works in time* $n^{\mathcal{O}(1)}$*, we shall say that the family is* $\mathrm{P}_{\mathrm{add}}$-uniform*.*

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

REMARK 2.11.

(i) Corollary 2.8 extends to all the polynomial hierarchy. That is, the power of real quantification is the same as that of digital quantification as long as the number of quantifier alternations is bounded. Surprisingly, if the number of quantifier alternations is not bounded, then the power of digital quantification is exactly $\mathrm{PAR}_{\mathrm{add}}$ and that of real quantification is at least additive exponential time, thus showing that the latter is more powerful than the former. For details see [4, 14].

(ii) In classical complexity theory, NP is a class of decision problems. Yet, if $S \in \mathsf{NP}$ and $x \in S$, a witness y for x can be computed in $\mathsf{FP}^{\mathsf{NP}}$ and thus in $\mathsf{FPSPACE}$

by computing its components one by one with an NP-routine. Looking at the proof of Corollary 2.8, we see that one can do the same with $\mathrm{NP}_{\mathrm{add}}$ and $\mathrm{FPAR}_{\mathrm{add}}$.

3. Counting classes. We now want to define counting classes, following the lines used in discrete complexity theory to define #P. This is the class of functions $f\colon \{0,1\}^\infty \to \mathbb{N}$ for which there exists an NP-machine M and a polynomial p such that, for all $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $f(x) = |\{y \in \{0,1\}^{p(n)} \mid M \text{ accepts } (x,y)\}|$. That is, $f(x)$ is the number of witnesses for x. A first remark is that over the reals, one can define two such complexity classes by counting the witnesses in an $\mathrm{NP}_{\mathrm{add}}$-machine, or in a $\mathrm{DNP}_{\mathrm{add}}$-machine, respectively.

Definition 3.1.

1. *We say that a function $f : \mathbb{R}^\infty \to \mathbb{N} \cup \{\infty\}$ belongs to the class $\#\mathrm{P}_{\mathrm{add}}$, if there exists a $\mathrm{NP}_{\mathrm{add}}$-machine M and a polynomial p such that, for all $n \in \mathbb{N}$, $x \in \mathbb{R}^n$,*

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

2. *We say that a function $f\colon \mathbb{R}^\infty \to \mathbb{N}$ belongs to the class $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$, if there exists a $\mathrm{DNP}_{\mathrm{add}}$-machine M and a polynomial p such that, for all $n \in \mathbb{N}$, $x \in \mathbb{R}^n$,*

$$f(x) = |\{y \in \{0,1\}^{p(n)} \mid M \text{ accepts } (x,y)\}|.$$

Remark 3.2.

(i) An unrestricted version of the class $\#\mathrm{P}_{\mathrm{add}}$ defined for machines over $\mathbb{R}$ which can multiply and divide, was defined by Meer in [27].

(ii) Note that it is not clear that $\mathrm{NP}_{\mathrm{add}} = \mathrm{DNP}_{\mathrm{add}}$ implies $\#\mathrm{P}_{\mathrm{add}} = \mathrm{D}\#\mathrm{P}_{\mathrm{add}}$, since now we are counting, not considering existence.

(iii) If f belongs to $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$, then the bit-size of $f(x)$ is bounded by a polynomial in the size of x. The same holds for $f \in \#\mathrm{P}_{\mathrm{add}}$ for those $x \in \mathbb{R}^n$ for which $f(x)$ is finite.

The next proposition locates the power of counting complexity classes within the landscape of known complexity classes. For interpreting the second inclusion, one should represent the value ∞ by some number in $\mathbb{R} \setminus \mathbb{N}$.

Proposition 3.3. *We have the following inclusions of complexity classes over $\mathbb{R}$*

$$\mathrm{D}\#\mathrm{P}_{\mathrm{add}} \subseteq \#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FPAR}_{\mathrm{add}}.$$

To prove Proposition 3.3 we will use the following result. Let $\mathrm{CINF}_{\mathrm{add}}$ be the problem to decide whether the solution set described by an additive circuit has infinitely many points.

Lemma 3.4. $\mathrm{CINF}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.*

Proof. Recall from §1.4 that the circuit satisfiability problem $\mathrm{CSAT}_{\mathrm{add}}$ is $\mathrm{NP}_{\mathrm{add}}$-complete. Adding a dummy variable to an additive circuit gives a (trivial) reduction from $\mathrm{CSAT}_{\mathrm{add}}$ to $\mathrm{CINF}_{\mathrm{add}}$, which shows the $\mathrm{NP}_{\mathrm{add}}$-hardness of $\mathrm{CINF}_{\mathrm{add}}$.

For the membership in $\mathrm{NP}_{\mathrm{add}}$, note that leaf sets are convex. So they are infinite if and only if they contain at least two points. Therefore, the following algorithm shows that membership of $\mathrm{CINF}_{\mathrm{add}}$ is in $\mathrm{NP}_{\mathrm{add}}$. On input $\mathcal{C}$ guess a leaf ν and guess $y^1, y^2 \in \mathbb{R}^n$. Then check whether $y^1 \neq y^2$ and whether y^1, y^2 reach the leaf ν. If yes, then accept, otherwise reject. □

Proof of Proposition 3.3. The first inclusion is clear. For the second, consider the algorithm that, in parallel, checks for each accepting leaf ν whether there is any point in D_ν and, if yes, whether D_ν has infinitely many points. These verifications

can be done in $\mathrm{NP}_{\mathrm{add}}$. If $|D_\nu| = \infty$ for some ν, then return ∞; else return the number of ν's such that $D_\nu \neq \emptyset$. This procedure clearly is in $\mathrm{FPAR}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}} = \mathrm{FPAR}_{\mathrm{add}}$. ∎

We will see in Theorem 4.7 below that the difference in power of $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ and $\#\mathrm{P}_{\mathrm{add}}$ is negligible.

We now focus on complete problems. To do so, we define the appropriate notions of reduction.

DEFINITION 3.5. *Let $f, g : \mathbb{R}^\infty \to \mathbb{N} \cup \{\infty\}$ and $\mathcal{C}$ be any of $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ or $\#\mathrm{P}_{\mathrm{add}}$.*

1. *We say that $\varphi : \mathbb{R}^\infty \to \mathbb{R}^\infty$ is a* parsimonious reduction *from f to g, if φ can be computed in polynomial time and, for all $x \in \mathbb{R}^\infty$, $f(x) = g(\varphi(x))$.*

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. *We say that a function g is* $\mathcal{C}$-hard *for 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 hardness and completeness with respect to Turing reductions are defined similarly.*

Let $\#\mathrm{CSAT}_{\mathrm{add}}$ denote the problem of counting the number of points of a semilinear set given by an additive circuit. (Note that this requires to compute a function with values in $\mathbb{N} \cup \{\infty\}$.)

THEOREM 3.6. *The counting problem $\#\mathrm{CSAT}_{\mathrm{add}}$ is $\#\mathrm{P}_{\mathrm{add}}$-complete.*

Proof. One just checks that the usual many-one reduction from the nondeterministic machine acceptance to additive circuit satisfiability is parsimonious. ∎

We close this section by recalling a principle introduced by Toda [38, 39], which allows to assign to any complexity class $\mathcal{C}$ of decision problems a corresponding counting complexity class $\# \cdot \mathcal{C}$.

DEFINITION 3.7. *Given a set $A \in \{0,1\}^\infty$ and a polynomial p, we define the function $\#_A^p : \{0,1\}^\infty \to \mathbb{N}$ which associates to $x \in \{0,1\}^n$ the number*

$$\#_A^p(x) = |\{y \in \{0,1\}^{p(n)} \mid (x,y) \in A\}|.$$

If $\mathcal{C} \subseteq 2^{\{0,1\}^\infty}$ is a complexity class of decision problems, then we define

$$\# \cdot \mathcal{C} = \{\#_A^p \mid A \in \mathcal{C} \text{ and } p \text{ a polynomial}\}.$$

Similarly, one assigns $\# \cdot \mathcal{C}$ and $\mathrm{D}\# \cdot \mathcal{C}$ to a complexity class $\mathcal{C}$ over $\mathbb{R}$.

Note that $\# \cdot \mathsf{P} = \#\mathsf{P}$, $\# \cdot \mathrm{P}_{\mathrm{add}} = \#\mathrm{P}_{\mathrm{add}}$, and $\mathrm{D}\# \cdot \mathrm{P}_{\mathrm{add}} = \mathrm{D}\#\mathrm{P}_{\mathrm{add}}$.

We will use the following important result due to Toda and Watanabe several times.

THEOREM 3.8 ([39]). *We have $\# \cdot \mathsf{PH} \subseteq \mathsf{FP}^{\#\mathsf{P}}$.*

4. Relationships between the real additive and discrete setting. The work of Koiran [24], Cucker and Koiran [14], and Fournier and Koiran [16, 17] establishes close relationships between the real additive and the discrete model of computation. Building on these techniques, we show in §4.1 that similar relationships hold for the counting classes. Then we use this in §4.2 to derive transfer theorems for counting classes between the additive real and the discrete setting.

4.1. The power of discrete oracles. The main result of this section, Theorem 4.1 stated below, says that for several classical complexity classes $\mathcal{C}$, the corresponding additive complexity classes $\mathcal{C}_{\mathrm{add}}$ are 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 an oracle in $\mathcal{C}$.

This result was stated and proved for the class $\mathrm{NP}_{\mathrm{add}}$ in [17]. Moreover, in [17, Remark 2] it was already 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 4.1 is the extension to the counting classes, and to the functional class $\mathrm{FPAR}_{\mathrm{add}}$.

THEOREM 4.1. *The following statements hold* ($k \geq 0$):

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

Observe that part two of this theorem implies that $\mathrm{D}\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}^{\#\mathsf{P}}_{\mathrm{add}}$.

As in Fournier and Koiran [16, 17], the proof of Theorem 4.1 relies on Meyer auf der Heide's (nonuniform) construction of small depth linear decision trees for point location in arrangements of hyperplanes [29, 30]. Before giving the proof, we need to develop some lemmas.

First, let us recall some terminology regarding arrangements of hyperplanes. For $s, n \in \mathbb{N}$ we define $\mathcal{H}_{s,n}$ to be the set of linear polynomials $a_0 + \sum_{i=1}^n a_i X_i$ with integer coefficients a_i such that $\sum_{i=0}^n |a_i| \leq 2^s$. 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*.

By Theorem 2.6, each face $F \in \mathcal{F}_{s,n}$ contains a rational point of bit-size at most $(sn)^c$, for some fixed constant $c > 0$. (Even though a face F may be described by a number of constraints exponential in n, s.) Therefore, $\log |\mathcal{F}_{s,n}| \leq (sn)^c$.

In what follows, we will encode a face $F \in \mathcal{F}_{s,n}$ by a triple $(s, n, x) \in \mathbb{N}^2 \times \mathbb{Q}^n$ such that $x \in F$ and the bit-size of x is at most $(sn)^c$. This way, we can describe all faces in $\mathcal{F}_{s,n}$, but of course, the description is not unique. Abusing notation, we will shortly express this by saying that the face F is represented by a "small rational point" x.

For a fixed polynomial t we define $\mathcal{H}_t$ as the union of the $\mathcal{H}_{t(n),n}$ over all $n \in \mathbb{N}$, and $\mathcal{F}_t$ as the union of the $\mathcal{F}_{t(n),n}$ over all $n \in \mathbb{N}$.

LEMMA 4.2. *Let M be an additive machine without real constants taking inputs in $\mathbb{R}^\infty \times \{0,1\}^\infty$ such that its running time is bounded by a polynomial t in the size of its first argument. The discrete relation*

$$R := \{(F, y) \in \mathcal{F}_t \times \{0,1\}^\infty \mid \forall x \in F \ \big(M \text{ accepts } (x, y)\big)\}$$

can be checked in P, that is, in classical polynomial time.

Proof. Running M on an input $(x, y) \in \mathbb{R}^n \times \{0,1\}^m$ takes at most $t(n)$ steps by assumption. On such an input, the machine M branches according to the signs of expressions $a_0 + \sum_{i=1}^n a_i x_i + \sum_{j=1}^m b_j y_j$, where $\sum_{i=0}^n |a_i| + \sum_{j=1}^m |b_j| \leq 2^{t(n)}$. Hence the hyperplane defined by $(a_0 + \sum_{j=1}^m b_j y_j) + \sum_{i=1}^n a_i X_i$ belongs to $\mathcal{H}_{t(n),n}$ for any $y \in \{0,1\}^m$. It follows that if x and x' belong to the same face F of $\mathcal{F}_{t(n),n}$, then for all y, (x, y) and (x', y) follow the same path in the decision tree induced by M. Therefore, if M accepts (x, y) for some $x \in F$, then it must accept (x, y) for all $x \in F$.

Therefore, checking that $(F, y) \in R$ can be done as follows. Let F be represented by the small rational point $x \in F$. Simulate the computation of the real machine M on input (x, y) by a Turing machine. Since M has no real constants and works in polynomial time, this simulation takes polynomial time (see [24, 4]). □

The following is an immediate consequence of Lemma 4.2 and the definition of Σ^k.

COROLLARY 4.3. *Let M be an additive machine as in Lemma 4.2, $k \in \mathbb{N}$, and $q_1, \ldots, q_k$ be polynomials. Consider the discrete relation*

$$R := \{(F, y) \in \mathcal{F}_t \times \{0,1\}^\infty \mid \forall x \in F \ \exists z_1 \forall z_2 \ldots Q z_k \\ \big(M \text{ accepts } (x, y, z_1, \ldots, z_k)\big)\},$$

where quantifiers alternate (Q is either existential or universal depending on the parity of k), the quantification is over $z_i \in \{0,1\}^{q_i(\mathrm{size}(x))}$. Then R can be checked in (classical) Σ^k.

Consider the following problem $\mathrm{FEVAL}_{\mathrm{add}}$: given a quantifier-free formula ψ of the first-order theory of $(\mathbb{R}, +, -, \leq)$ with k free variables and a point $x \in \mathbb{R}^k$, decide whether $\psi(x)$ holds. Note that the formula ψ can be encoded as an element of $\mathbb{R}^\infty$ in a straightforward way. In the following, we will identify ψ with its encoding. It is well-known that $\mathrm{FEVAL}_{\mathrm{add}} \in \mathrm{P}^0_{\mathrm{add}}$.

The next result is proved similarly as Lemma 4.2.

LEMMA 4.4. *Let M be a machine solving $\mathrm{FEVAL}_{\mathrm{add}}$ in time bounded by a polynomial t in $\mathrm{size}(\psi)$. Then the following set belongs to* PSPACE:

$$L := \{F \in \mathcal{F}_t \mid \forall \psi \in F \ Q_1 z_1 Q_2 z_2 \ldots Q_n z_n \ \big(M \text{ accepts } (\psi, z_1, \ldots, z_n)\big)\}.$$

Here $Q_i \in \{\forall, \exists\}$, $z_i \in \{0,1\}$. Moreover, n denotes the size of ψ. Hence the number of free variables of ψ is at most n and the behavior of M on $(\psi, z_1, \ldots, z_n)$ is well-defined.

Given a polynomial t, the *point location problem* for t is the problem of computing, for a given input $x \in \mathbb{R}^\infty$, a small rational point of the uniquely determined face $F \in \mathcal{F}_{t(\mathrm{size}(x)),\mathrm{size}(x)}$ in which x lies. The following crucial statement is proved by Fournier and Koiran [17, Theorem 2].

PROPOSITION 4.5. *For any polynomial t, the point location problem can be solved in $(\mathrm{FP}^0_{\mathrm{add}})^{\mathsf{NP}}$. That is, a small rational point of the face $F \in \mathcal{F}_{t(n),n}$ containing the input point can be computed in polynomial time by an additive machine using a classical oracle in* NP.

We remark that in [17], a face F is represented by a system $\mathcal{S}$ of $n^{\mathcal{O}(1)}$ linear inequalities with integer coefficients of bit-size $n^{\mathcal{O}(1)}$, such that the polyhedral set defined by the system $\mathcal{S}$ is non-empty and is contained in the face F. However, note that since linear programming (discrete setting) is in polyomial time [21, 22], we can always compute from the system $\mathcal{S}$ a small rational point of F in polynomial time.

Proof of Theorem 4.1. 1. Assume that $A \in \Sigma^k_{\mathrm{add}}$. Then (cf. Remark 2.11(i)) there exist polynomials $q_1, \ldots, q_k$ and $B \in \mathrm{P}_{\mathrm{add}}$ such that for all $x \in \mathbb{R}^\infty$

$$x \in A \iff \exists z_1 \forall z_2 \ldots Q z_k \ (x, z) \in B,$$

where quantifiers alternate (Q is either existential or universal depending on the parity of k) and the quantification is over $z_i \in \{0,1\}^{q_i(\mathrm{size}(x))}$.

Let M_B be an additive machine deciding B in time bounded by some polynomial t and $\alpha_1, \ldots, \alpha_\ell$ be the constants occurring in the program of M_B other than 0 or 1.

Denote $\alpha = (\alpha_1, \ldots, \alpha_\ell)$ and let

$$C = \{(x, z, v) \in \mathbb{R}^\infty \times \{0,1\}^\infty \times \mathbb{R}^\ell \mid M_B \text{ accepts } (x, z) \text{ when replacing } \alpha \text{ by } v \text{ in its program}\}.$$

Clearly, $C \in \mathrm{P}^0_{\mathrm{add}}$ and, for all $(x, z) \in \mathbb{R}^\infty \times \{0,1\}^\infty$, $(x, z) \in B \iff (x, z, \alpha) \in C$. In order to prove that $A \in \mathrm{P}^{\Sigma^k}_{\mathrm{add}}$, it is sufficient to show that the set

$$A' = \{(x, v) \in \mathbb{R}^\infty \times \mathbb{R}^\ell \mid \exists z_1 \forall z_2 \ldots Q z_k \ (x, z, v) \in C\}$$

belongs to the class $(\mathrm{P}^0_{\mathrm{add}})^{\Sigma^k}$. This reasoning shows that we may assume without loss of generality that M_B has no real constants, i.e., that $B \in \mathrm{P}^0_{\mathrm{add}}$.

Since $B \in \mathrm{P}^0_{\mathrm{add}}$, we may use Corollary 4.3 (used without the y in the input) to deduce that the discrete language

$$R = \{F \in \mathcal{F}_t \mid \forall x \in F \ \exists z_1 \forall z_2 \ldots Q z_k \ \big(M_B \text{ accepts } (x, z)\big)\}$$

lies in the class Σ^k. (Recall that t bounds the running time of M_B.) Consider now the following algorithm. On input $x \in \mathbb{R}^n$ locate x in a face F of $\mathcal{F}_{t(n),n}$ (due to Proposition 4.5, this can be done in $(\mathrm{FP}^0_{\mathrm{add}})^{\mathsf{NP}}$). Then decide whether $F \in R$ by an oracle call to Σ^k. This algorithm works in $\mathrm{P}^{\Sigma^k}_{\mathrm{add}}$ and decides A.

The above reasoning shows that $A \in \mathrm{P}^{\Sigma^k}_{\mathrm{add}}$. This immediately implies the inclusions for Π^k and PH. And, since $\mathrm{PH}_{\mathrm{add}}$ is closed under Turing reductions, we even get equality in this case.

2. Let $\varphi\colon \mathbb{R}^\infty \to \mathbb{N}$ be a counting problem in $\mathrm{D}\#\cdot\Sigma^k_{\mathrm{add}}$. Then there exist $B \in \mathrm{P}_{\mathrm{add}}$ and polynomials $p, q_1, \ldots, q_k$ such that for all $n \in \mathbb{N}$ and all $x \in \mathbb{R}^n$

$$\varphi(x) = |\{y \in \{0,1\}^{p(n)} \mid \exists z_1 \forall z_2 \ldots Q z_k \ (x, y, z_1, \ldots, z_k) \in B\}|$$

where quantifiers alternate and the quantification is over $z_i \in \{0,1\}^{q_i(\mathrm{size}(x))}$.

Let M_B be some additive machine deciding B in time t for some polynomial t. As in the proof of part one, we can assume that M_B has no real constants. By Corollary 4.3, the map $\psi\colon \mathcal{F}_t \to \mathbb{N}$ defined for $F \in \mathcal{F}_{t(n),n}$ by

$$\psi(F) := |\{y \in \{0,1\}^{p(n)} \mid \forall x \in F \ \exists z_1 \forall z_2 \ldots Q z_k \ (x, y, z_1, \ldots, z_k) \in B\}|$$

lies in the discrete counting class $\# \cdot \Sigma^k$. Note that $\varphi(x) = \psi(F)$ for $F \in \mathcal{F}_{t(n),n}$, $x \in F$.

Consider now the following algorithm. On input $x \in \mathbb{R}^n$, locate x in a face F of $\mathcal{F}_{t(n),n}$. Then compute $\psi(F)$ by an oracle call to $\# \cdot \Sigma^k$ and return $\psi(F)$. This algorithm works in $\mathrm{FP}^{\#\cdot\Sigma^k}_{\mathrm{add}}$ and computes $\varphi(x)$.

The above proves the inclusion $\mathrm{D}\# \cdot \Sigma^k_{\mathrm{add}} \subseteq \mathrm{FP}^{\#\cdot\Sigma^k}_{\mathrm{add}}$. The inclusion $\mathrm{D}\# \cdot \Pi^k_{\mathrm{add}} \subseteq \mathrm{FP}^{\#\cdot\Pi^k}_{\mathrm{add}}$ is proved similarly and it follows that $\mathrm{D}\# \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}^{\#\cdot\mathsf{PH}}_{\mathrm{add}}$.

3. The inclusion $\mathrm{P}^{\mathsf{PSPACE}}_{\mathrm{add}} \subseteq \mathrm{PAR}_{\mathrm{add}}$ is clear, since $\mathrm{PAR}_{\mathrm{add}}$ is closed under Turing reductions. Consider the subset DTRAO of the theory of the reals with addition and order which consists of the sentences all of whose variables z satisfy a constraint of the form $z = 0 \vee z = 1$. In [14] it is proven that DTRAO is $\mathrm{PAR}_{\mathrm{add}}$-complete. It is therefore sufficient to show that $\mathrm{DTRAO} \in \mathrm{P}^{\mathsf{PSPACE}}_{\mathrm{add}}$.

Consider now the following algorithm. On input a digitally quantified first-order sentence $\varphi = Q_1 z_1 Q_2 z_2 \ldots Q_k z_k \ \psi(z_1, \ldots, z_k)$, of the theory of $(\mathbb{R}, +, -, \leq)$, where ψ

is quantifier free of size n, we locate ψ in a face $F \in \mathcal{F}_{t(n),n}$, where n is the size of ψ. By construction, φ is true iff F belongs to the set L in Lemma 4.4. We then decide this membership by an oracle call to PSPACE. This algorithm decides $\mathrm{DTRAO} \in \mathrm{P}_{\mathrm{add}}^{\mathsf{PSPACE}}$.

4. The equality $\mathrm{FP}_{\mathrm{add}}^{\mathrm{PAR}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\mathsf{PSPACE}}$ follows from the third statement of Theorem 4.1. To prove the first equality, let $f \in \mathrm{FPAR}_{\mathrm{add}}$, $x \in \mathbb{R}^n$, and $y = f(x) \in \mathbb{R}^{p(n)}$. Let $\alpha_1, \ldots, \alpha_k$ be the constants occurring in the additive machine that generates the circuits computing f as described in Definition 2.10. For $\ell \leq p(n)$ we have

$$y_\ell = \sum_{i=1}^{n} u_i^{(\ell)} x_i + \sum_{j=1}^{k} v_j^{(\ell)} \alpha_j + b^{(\ell)} \tag{4.1}$$

where $u_i^{(\ell)}, v_j^{(\ell)}$ and $b^{(\ell)}$ are integers of bit-size at most $q(n)$ for a polynomial q. Let B_x be the relation defined by

$$B_x = \{(s, i, \ell, x) \in \mathbb{N}^3 \times \mathbb{R}^\infty \mid \text{ the } s\text{-th bit of } u_i^{(\ell)} \text{ is } 1\}$$

and B_α, B_1 the analogous relations for $v_j^{(\ell)}$ and $b^{(\ell)}$. We claim that $B_x, B_\alpha, B_1 \in \mathrm{PAR}_{\mathrm{add}}$. In fact, the parallel algorithm deciding any of these relations simulates the behaviour of $\mathcal{C}_n$ (the circuit computing the restriction of f to $\mathbb{R}^n$) on input x keeping expressions in the form (4.1) instead of actually performing the arithmetic operations.

To compute $f(x)$ in $\mathrm{FP}_{\mathrm{add}}^{\mathrm{PAR}_{\mathrm{add}}}$, one uses the oracles B_x, B_α, and B_1 to obtain the binary expansions of $u_i^{(\ell)}$, $v_j^{(\ell)}$ and $b^{(\ell)}$, for all ℓ, i, j. Then compute y by (4.1). □

The following corollary is a real analogue of Toda and Watanabe's Theorem 3.8.

COROLLARY 4.6. *We have* $\mathrm{D}\# \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$.

Proof. We conclude from Theorem 4.1(2) and Theorem 3.8 that

$$\mathrm{D}\# \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\cdot\mathsf{PH}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\mathsf{FP}^{\#\mathsf{P}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$$

which proves the claim. □

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

THEOREM 4.7. *We have* $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\mathrm{D}\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$.

Proof. Clearly, it is enough to show that $\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$. For this, it is sufficient to prove that

$$\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\mathrm{D}\#\cdot\mathrm{NP}_{\mathrm{add}}}. \tag{4.2}$$

Indeed, by Corollary 4.6, we know that $\mathrm{D}\# \cdot \mathrm{NP}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$.

In order to prove (4.2), let $\varphi \in \#\mathrm{P}_{\mathrm{add}}$. Then there exist a polynomial p and an additive machine M working in polynomial time such that, for all $n \in \mathbb{N}$, $x \in \mathbb{R}^n$,

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

Using Lemma 3.4 we can find out in $\mathrm{NP}_{\mathrm{add}}$ whether $\varphi(x)$ is infinite on input $x \in \mathbb{R}^n$. Assume then that $\varphi(x)$ is finite. Since leaf sets are convex, they are either infinite or consist of just one point. In the case that $\varphi(x)$ is finite, we have

$$\varphi(x) = |\{\nu \in \{0,1\}^{t(n)} \mid \exists y \in \mathbb{R}^{p(n)} \text{ input } (x, y) \text{ reaches the accepting leaf } \nu\}|,$$

since y is uniquely determined. Define $B := \cup_{n\in\mathbb{N}} B_n$, where

$$B_n := \{(x,\nu) \in \mathbb{R}^n \times \{0,1\}^{t(n)} \mid \exists y \in \mathbb{R}^{p(n)} \text{ input } (x,y) \text{ reaches the accepting leaf } \nu\}.$$

Then we have $B \in \mathrm{NP}_{\mathrm{add}}$. This implies that $\varphi \in \mathrm{D}\# \cdot \mathrm{NP}_{\mathrm{add}}$. □

REMARK 4.8.

(i) Statements analogous to Theorem 4.1 hold in the constant-free setting, e.g.,

$$\mathrm{NP}^0_{\mathrm{add}} \subseteq (\mathrm{P}^0_{\mathrm{add}})^{\mathsf{NP}}, \quad \mathrm{D}\#\mathrm{P}^0_{\mathrm{add}} \subseteq (\mathrm{FP}^0_{\mathrm{add}})^{\#\mathsf{P}}, \quad \mathrm{FPAR}^0_{\mathrm{add}} = (\mathrm{FP}^0_{\mathrm{add}})^{\mathsf{PSPACE}}.$$

(ii) Similarly to Theorem 4.7, we have $\mathrm{FP}^{\#\Sigma^k_{\mathrm{add}}}_{\mathrm{add}} = \mathrm{FP}^{\mathrm{D}\#\cdot\Sigma^k_{\mathrm{add}}}_{\mathrm{add}} = \mathrm{FP}^{\#\mathsf{P}}_{\mathrm{add}}$ for $k \geq 0$.

4.2. Boolean parts and transfer theorems. A problem that has attracted much attention in real complexity is the computation of Boolean parts [8, 12, 13, 14, 24, 25]. Roughly speaking, this amounts to characterizing, in terms of classical complexity classes, the power of resource bounded machines over $\mathbb{R}$ when their inputs are restricted to be binary. In this section, we will be interested in the Boolean parts of counting classes.

DEFINITION 4.9. *Let $\mathcal{C}$ be a counting class over $\mathbb{R}$. Its* Boolean part *is the classical complexity class* $\mathrm{BP}(\mathcal{C}) = \{f : \{0,1\}^\infty \to \mathbb{N} \mid f = g_{|\{0,1\}^\infty} \text{ for some } g \in \mathcal{C}\}$.

PROPOSITION 4.10. *We have* $\mathrm{BP}(\mathrm{D}\#\mathrm{P}_{\mathrm{add}}) = \#\mathsf{P}/\mathsf{poly}$.

Proof. The proof closely follows that of [4, Theorem 2, Chapter 22]. Consider a function f in $\#\mathsf{P}/\mathsf{poly}$. There is a polynomial q and an advice function h such that $h(n)$ belongs to $\{0,1\}^{q(n)}$ for all n. Furthermore, there are an NP-machine M and a polynomial p such that M accepts for exactly $f(x)$ witnesses in $\{0,1\}^{p(n)}$ on input $(x,h(n))$, for all $x \in \{0,1\}^n$. Let us code in a single number $\xi_h \in \mathbb{R}$ the sequence of advices $h(1), h(2), \ldots$. Then we can consider a $\mathrm{DNP}_{\mathrm{add}}$-machine which, for each input $x \in \{0,1\}^n$, first produces the digits of ξ_h and obtains $h(n)$, and then simulates M on $(x,h(n))$. This shows that $\#\mathsf{P}/\mathsf{poly} \subseteq \mathrm{BP}(\mathrm{D}\#\mathrm{P}_{\mathrm{add}})$.

Conversely, let us consider a function f in the Boolean part of $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ defined by a $\mathrm{DNP}_{\mathrm{add}}$-machine M with time bound q. The computation of M on inputs of size n is described by a linear decision tree T of depth $q(n)$. Therefore, if $\alpha_1, \ldots, \alpha_k$ are the real constants of M then, for each $x \in \{0,1\}^n$, the test performed by T at a node i has the form $g_i(x,\alpha) \geq 0$ with

$$g_i(x,\alpha) = \sum_{j=1}^{n} a_{ij}x_j + \sum_{j=1}^{k} b_{ij}\alpha_j + c_i, \tag{4.3}$$

and where a_{ij}, b_{ij} and c_i are integers of bit-size polynomial in n. For a given $x \in \{0,1\}^n$, according to the outcome of the test (4.3), the point $\alpha \in \mathbb{R}^k$ satisfies then an inequality of the form $g_{i,x}(\alpha) \geq 0$ or $g_{i,x}(\alpha) < 0$ where $g_{i,x} \in \mathbb{Z}[Y_1, \ldots, Y_k]$ is defined by $g_{i,x}(y) = g_i(x,y)$. Let Φ be the system of all these linear inequalities, for i varying over all branching nodes of T and x varying over the 2^n possible points of $\{0,1\}^n$. The system Φ is satisfied by α. Then, according to Theorem 2.6, there is a point $\beta_n \in \mathbb{Q}^k$ all of whose coordinates have polynomial bit-size in n, which also satisfies Φ. Thus, if we replace $\alpha = (\alpha_1, \ldots, \alpha_k)$ by β_n in the tree T, the path followed by any $x \in \{0,1\}^n$ will not change and x will be accepted or rejected as in T.

Let us now consider the function $h : \mathbb{N} \to \mathbb{Q}^k$ defined by $h(n) = \beta_n$. Since the bit-size of β_n is polynomial in n we may interpret h as an advice function. The classical machine which, with input $(x, h(\mathrm{size}(x)))$, simulates the behavior of M with the constants $\alpha_1, \ldots, \alpha_k$ replaced by $h(\mathrm{size}(x))$ shows that $f \in \#\mathsf{P}/\mathsf{poly}$. □

Combining Proposition 4.10 with the results of §4.1, we obtain:

COROLLARY 4.11. *We have the following transfer results:*

1. $\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}$ *iff* $\mathrm{D}\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}$ *iff* $\#\mathsf{P} \subseteq \mathsf{FP/poly}$.
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.*

Proof. The first equivalence of (1) follows from Theorem 4.7. In the second equivalence of (1), the direction from left to right follows from

$$\#\mathsf{P} \subseteq \#\mathsf{P/poly} = \mathrm{BP}(\mathrm{D}\#\mathrm{P}_{\mathrm{add}}) \subseteq \mathrm{BP}(\mathrm{FP}_{\mathrm{add}}) \subseteq \mathsf{FP/poly}.$$

Here, the equality holds by Proposition 4.10, the last inclusion holds by Remark 4.12 below, and the one before the last is true by assumption. For the other direction, we use Theorem 4.1(2) to get $\mathrm{D}\#\mathrm{P}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\mathsf{FP/poly}} = \mathrm{FP}_{\mathrm{add}}$, where the second inclusion follows by assumption. The other equivalences can be shown similarly. □

REMARK 4.12. One can extend the definition of Boolean parts to classes of (not necessarily counting) functions over the reals and consider, for instance,

$$\mathrm{BP}(\mathrm{FP}_{\mathrm{add}}) = \{f : \{0,1\}^\infty \to \{0,1\}^\infty \mid f = g_{|\{0,1\}^\infty} \text{ for some } g \in \mathrm{FP}_{\mathrm{add}}\}.$$

One can then use the same arguments to show that

$$\mathrm{BP}(\mathrm{FP}_{\mathrm{add}}) = \mathsf{FP/poly},\ \mathrm{BP}(\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}) = \mathsf{FP}^{\#\mathsf{P}}\mathsf{/poly},\ \mathrm{BP}(\mathrm{FPAR}_{\mathrm{add}}) = \mathsf{FPSPACE/poly}.$$

Also, for machines without constants we have the following, easier to prove, results:

$$\mathrm{BP}(\mathrm{FP}_{\mathrm{add}}^0) = \mathsf{FP},\ \mathrm{BP}((\mathrm{FP}_{\mathrm{add}}^0)^{\#\mathrm{P}_{\mathrm{add}}^0}) = \mathsf{FP}^{\#\mathsf{P}},\ \mathrm{BP}(\mathrm{FPAR}_{\mathrm{add}}^0) = \mathsf{FPSPACE}.$$

4.3. Complete counting problems. We obtain from Theorem 4.1 the following result, providing us with plenty of complete problems for the class $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$.

PROPOSITION 4.13. *Let $f\colon \mathbb{R}^\infty \to \mathbb{N}$ belong to $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$ and assume that the restriction of f to $\mathbb{Z}^\infty$ is $\#\mathsf{P}$-complete with respect to Turing reductions. Then f is $\#\mathrm{P}_{\mathrm{add}}$-complete and thus $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-complete with respect to Turing reductions.*

Proof. This is an immediate consequence of Theorem 4.7. □

Proposition 4.13 yields plenty of Turing complete problems in $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$. 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 [41, 42] 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 4.13, this immediately implies the following corollary.

COROLLARY 4.14. *The problems $\#\mathrm{TSP}_{\mathbb{R}}$ and $\#\mathrm{PM}_{\mathbb{R}}$ are $\mathrm{D}\#\mathrm{P}_{\mathrm{add}}$-complete with respect to Turing reductions.*

Note that the problem to count the number of perfect matchings of a given bipartite graph is equivalent to the famous problem to evaluate the *permanent* of a matrix with entries in $\{0,1\}$.

REMARK 4.15.

1. Results for $\mathrm{PAR}_{\mathrm{add}}$ and $\mathrm{FPAR}_{\mathrm{add}}$ similar to Proposition 4.13 follow from Theorem 4.1 in the same manner.

2. There is an algebraic theory of NP-completeness due to Valiant [7, 40, 43], which captures the complexity to evaluate the generating functions of graph properties. For instance, the generating function of the property "perfect matching" is the permanent of a real matrix, which turns out to be complete in this theory. The functional problems studied in this theory take real values on real inputs and thus differ substantially from the counting problems studied in this paper. It would be interesting to clarify the relationship between these two approaches.

5. Complexity to compute topological invariants. In the computational problems studied in this section, it is always assumed that the input is an additive circuit $\mathcal{C}$ and X is the semi-linear set accepted by $\mathcal{C}$.

In the following subsections, we characterize the complexity to compute several basic invariants of a semi-linear set X. These invariants are the dimension (§5.1), the number of connected components (§5.2), the Euler characteristic (§5.3.4), and the Betti numbers (§5.3.5). In §6 we show that corresponding completeness results for the Turing model hold as well.

5.1. Complexity of computing 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 non-emptyness using the dimension function.

THEOREM 5.1. *For all $d \geq 0$, the problem $\mathrm{DIM}_{\mathrm{add}}(d)$ is $\mathrm{NP}_{\mathrm{add}}$-complete.*

The main tool in proving Theorem 5.1 is the following proposition. It states that if a polyhedron P has dimension at least d, then there is a projection of P onto a d-dimensional coordinate subspace containing a scaled down d-dimensional standard simplex. This can be used to construct an $\mathrm{NP}_{\mathrm{add}}$-certificate.

To formally state this proposition we define $e_j = (0,\ldots,0,1,0,\ldots,0)^{\mathrm{T}} \in \mathbb{R}^n$, the column unit vector with 1 in the jth place. Also, recall that we denote by $[m]$ the set $\{1,\ldots,m\}$, for any $m \in \mathbb{N}$.

PROPOSITION 5.2. *Let P be a polyhedron in $\mathbb{R}^n$ of dimension $d \geq 0$ defined by a system $A_1 x \leq b_1$; $A_2 x < b_2$, where $A_1 \in \mathbb{R}^{N_1 \times n}, A_2 \in \mathbb{R}^{N_2 \times n}, b_1 \in \mathbb{R}^{N_1}$ and $b_2 \in \mathbb{R}^{N_2}$. Then there exist points $x^{(0)},\ldots,x^{(d)} \in P$, $\rho > 0$, and an injective map $\pi\colon [d] \to [n]$ such that for all $\ell, i \in [d]$ we have $x^{(\ell)}_{\pi(i)} - x^{(0)}_{\pi(i)} = \rho\delta_{\ell,i}$. Here δ is the Kronecker's delta.*

Proof. We first prove the result for the case $d = n$. Let $N = N_1 + N_2$, $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \in \mathbb{R}^{N\times n}$, $b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \in \mathbb{R}^N$, and denote by $a_i^{\mathrm{T}} \in \mathbb{R}^n$ the ith row of $A = (a_{i,j})$. W.l.o.g. $A \neq 0$. Since $d = n$, the polyhedron

$$P^* := \{x \in \mathbb{R}^n \mid Ax < b\}.$$

is non-empty. Let $x^* \in P^*$ and put $y^* := b - Ax^* > 0$, $\varepsilon := \min_{i\leq N} y_i^* = y_{i_0}^*$. Then $a_i^{\mathrm{T}} x^* \leq b_i - \varepsilon$ for all $i \leq N$. Define $\rho = \varepsilon/(2\max_{i,j}|a_{ij}|)$. For all i,j we have $\rho|a_i^{\mathrm{T}} e_j| = \rho|a_{ij}| \leq \varepsilon/2$. This implies that, for all i,

$$|a_i^{\mathrm{T}}(x^* + \rho e_j)| \leq b_i - \varepsilon + \frac{\varepsilon}{2} < b_i.$$

The points $x^{(0)} = x^*$, $x^{(j)} = x^* + \rho e_j$ together with the identity map $\pi\colon [n] \to [n]$ satisfy the statement.

We now consider the general case, with $d \geq 1$ (the case $d = 0$ is trivial). Since $\dim P = d$, there exists $I \subseteq [N_1]$ with $|I| \geq n - d$ such that the set

$$P^* := \{x \in \mathbb{R}^n \mid a_i^{\mathrm{T}} x = b_i \text{ for } i \in I, \text{ and } a_i^{\mathrm{T}} x < b_i \text{ for } i \notin I\}$$

is included in P and has dimension d. After eliminating redundant equalities if necessary, we can assume $|I| = n - d$.

In what follows, if $x \in \mathbb{R}^n$ and $J \subseteq [n]$, we denote by x_J the vector obtained by removing from x the coordinates with index not in J. Write $\bar{J} = [n] - J$. Let H be the affine linear variety given by $a_i^{\mathrm{T}} x = b_i$ for $i \in I$. Since $\dim H = d$ there exists $J \subseteq [n]$, $|J| = d$, such that we can express the coordinates x_j, $j \notin J$, in terms of the coordinates x_j, $j \in J$. More precisely, there exist $S \in \mathbb{R}^{\bar{J} \times J}$ and $c \in \mathbb{R}^{\bar{J}}$ such that

$$x_{\bar{J}} = S x_J + c. \tag{5.1}$$

The projection of P^* on $\mathbb{R}^J$ has dimension d. It is given by the set of strict inequalities obtained by substituting $x_{\bar{J}}$ in the inequalities $a_i x < b_i$, $i \notin I$ according to (5.1).

We now apply the full dimensional case to this set of inequalities to find points $x_J^{(\ell)} \in \mathbb{R}^J$, $\ell = 0, \ldots, d$, satisfying the statement in $\mathbb{R}^J$ (for some bijection $\pi\colon [d] \to J$). Finally, we lift these points to $x^{(\ell)} \in \mathbb{R}^n$, $\ell \in [d]$, by using (5.1). □

Proof of Theorem 5.1. The hardness is immediate since adding d dummy variables reduces circuit satisfiability $\mathrm{CSAT}_{\mathrm{add}}$ to $\mathrm{DIM}_{\mathrm{add}}(d)$. (Here we use the convention $\dim \emptyset = -1$.)

For the membership, note that leaf sets are convex. Therefore, if such a leaf set contains the vertices of a d-dimensional simplex in $\mathbb{R}^n$, it contains the whole simplex. This ensures the correctness of the following algorithm for $\mathrm{DIM}_{\mathrm{add}}(d)$.

input $\mathcal{C}$
`compute` n`, the number of input gates of` $\mathcal{C}$
`guess an accepting leaf` ν`,` $x^{(0)}, \ldots, x^{(d)} \in \mathbb{R}^n$`,` $\rho \in \mathbb{R}$`, and` $\pi\colon [d] \to [n]$
`check` whether $x^{(0)}, \ldots, x^{(d)}$, ρ and π satisfy the statement of Proposition 5.2
if $x^{(0)}, \ldots, x^{(d)}$ reach the leaf ν **then** `ACCEPT` **else** `REJECT`

This is clearly an $\mathrm{NP}_{\mathrm{add}}$-algorithm. □

5.2. 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 $\mathcal{C}$.

The main result of this section is the following.

THEOREM 5.3. *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.*

This result is inspired by an early paper by Reif [35] (see also [36]), 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. However, we borrow from Reif's proof the idea to describe PSPACE by symmetric polynomial space Turing machines, see §5.2.1.

The problem $\mathrm{REACH}_{\mathrm{add}}$ has a similar flavour as the undirected reachability problem for succinctly represented graphs, which asks whether two nodes s, t of such a graph G are connected by a path. By a *succinct representation* of a graph [18] we understand a Boolean circuit which decides, for a pair of nodes given in binary encoding, whether they are adjacent. This representation allows a polynomial size representation of graphs with exponentially many nodes. It is known that the undirected reachability problem for succinctly represented graphs is PSPACE-complete. A detailed treatment of the complexity of succinct problems can be found in [1].

The rest of this section is devoted to the proof of Theorem 5.3. It is organized as follows: after two preparatory subsections on symmetric Turing machines and embedding graphs, we provide the lower bound part of the proof in §5.2.3. The upper bound part of the proof is given in §5.2.4.

5.2.1. Symmetric Turing machines. Roughly speaking, a symmetric Turing machine [26] 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 $\mathrm{REACH}_{\mathrm{add}}$.

We briefly recall the notions which are essential for our proof. A (one tape) *symmetric Turing machine* M is given by $(Q, \Sigma, \Sigma_0, s, t, \Delta)$, where Q is a finite state space, $s \in Q$ is the initial state, $t \in Q$ is the final state, and Δ is a finite set of transitions. We will assume that the input alphabet Σ_0 equals $\{0, 1\}$ and that the machine alphabet Σ contains $0, 1$ and the blank '♭'. The transitions $\delta \in \Delta$ are either of the form $\delta_1 = (p, a, 0, b, q)$ or $\delta_2 = (p, ab, cd, q)$, where $p, q \in Q$ and $a, b, c, d \in \Sigma$. The transition δ_1 is to be interpreted as follows: if the current state of the machine is p and the head of the machine is above a cell containing the symbol a, then the machine may rewrite this symbol by b and enter the state q without moving the head. Similarly, reading backwards, if the machine is in state q and the head of the machine is above a cell containing the symbol b, then the machine may rewrite this symbol by a and enter the state p without moving the head. The interpretation of the transition δ_2 is as follows: If the current state is p, the head of the machine is above a cell containing a and the symbol in the cell to the immediate right is b, then the the machine may rewrite a by c and b by d, move one step to the right and enter the state q. The transition δ_2 may also be read backwards with the obvious interpretation. A *configuration* of M is an element $(q, h, w) \in Q \times \mathbb{N} \times \Sigma^{\mathbb{N}}$, where q is the current state, h the position of the head and w the current tape contents. (All but finitely many components of w are blanks.) The transitions induce a symmetric relation on the set of configurations, which defines the (undirected) configuration graph.

In [26, Thm. 1] it is shown that any language recognized by a deterministic Turing machine may be recognized by a symmetric Turing machine respecting the same space bound.

5.2.2. Embedding graphs by straight-line segments. We first define two types of products of graphs. Then we study embeddings of such graph products in Euclidean space such that point location in these embedded graphs can be done by additive circuits. This will be needed for the lower bound proof in §5.2.3.

DEFINITION 5.4. *Let $G_i = (V_i, E_i)$ be graphs, $1 \leq i \leq t$.*

1. *The* product *$G_1 \times \cdots \times G_t$ is defined as a graph with the set of nodes $V_1 \times \cdots \times V_t$; two distinct nodes $u = (u_1, \ldots, u_t)$ and $v = (v_1, \ldots, v_t)$ are adjacent iff there exists i such that $\{u_i, v_i\} \in E_i$ and $u_j = v_j$ for all $j \neq i$. If $G_i = G$ for all i we will write G^t instead of $G \times \cdots \times G$.*

2. *The* extended product $G_1 \otimes \cdots \otimes G_t$ *also has set of nodes* $V_1 \times \cdots \times V_t$. *Now two distinct nodes* $u = (u_1, \ldots, u_t)$ *and* $v = (v_1, \ldots, v_t)$ *are adjacent iff for all* i *either* $u_i = v_i$ *or* $\{u_i, v_i\} \in E_i$.

Let $G = (V, E)$ be a graph and $\varphi\colon V \to \mathbb{R}^n$ be an injective map. We assign to each edge $e = \{u, v\}$ the closed (straight-line) segment $\varphi(e) := \{t\varphi(u) + (1-t)\varphi(v) \mid t \in [0,1]\}$.

DEFINITION 5.5. *The injective map* $\varphi\colon V \to \mathbb{R}^n$ induces an embedding *of the graph* $G = (V, E)$ *into* $\mathbb{R}^n$ (*by straight-line segments*) *iff*

$$\forall v \in V\ \forall e, e' \in E\ \big(\varphi(v) \in \varphi(e) \Longrightarrow v \in e\big) \wedge \big(\varphi(e) \cap \varphi(e') \neq \emptyset \Longrightarrow e \cap e' \neq \emptyset\big).$$

The edge skeleton *of the embedding is defined as the union of the segments corresponding to all edges.*

We denote by K_m the complete graph on the set of nodes $[m]$ and embed it in $\mathbb{R}^m$ by sending the node i to the ith canonical basis vector e_i. Thus we interpret K_m as the standard simplex in $\mathbb{R}^m$.

LEMMA 5.6.

1. *Assume that* $\varphi_i\colon V_i \to \mathbb{R}^{n_i}$ *induces an embedding of* G_i *for* $i \in [t]$. *Then* $V_1 \times \cdots \times V_t \to \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_t}, (v_1 \ldots, v_t) \mapsto (\varphi_1(v_1), \ldots, \varphi_t(v_t))$ *induces an embedding of* $G_1 \times \cdots \times G_t$.

2. *Let* $\varphi\colon V \to \mathbb{R}^n - \{0\}$ *induce an embedding of* $G = (V, E)$ *and* $m \in \mathbb{N}, m > 0$. *Assume further that* $\varphi(u), \varphi(v)$ *are linearly independent for all edges* $\{u, v\} \in E$. *Define* $\psi\colon [m] \times V \to (\mathbb{R}^n)^m$ *by* $\psi(i, v) := (0, \ldots, 0, \varphi(v), 0, \ldots, 0)$ *with* 0 *everywhere except at position* i. *Then* ψ *induces an embedding of the extended product* $K_m \otimes G$ *into* $(\mathbb{R}^n)^m$.

Proof. 1. This can be shown by straightforward verification.

2. The proof is a bit cumbersome, since it requires several case distinctions. Assume for simplicity $m = 2$. Let $e = \{(1, u), (2, v)\}$, $e' = \{(1, u'), (2, v')\}$ be edges of $K_2 \otimes G$ such that $\psi(e) \cap \psi(e') \neq \emptyset$. Hence there exist $s, t \in [0, 1]$ such that $t\psi(1, u) + (1-t)\psi(2, v) = s\psi(1, u') + (1-s)\psi(2, v')$. This means that $t\varphi(u) = s\varphi(u')$ and $(1-t)\varphi(v) = (1-s)\varphi(v')$. We claim that $e \cap e' \neq \emptyset$.

If $t = 0$, then $s = 0$ (since $\mathrm{im}\varphi \subseteq \mathbb{R}^n - \{0\}$), hence $\varphi(v) = \varphi(v')$. Therefore $v = v'$ and we are done. Similarly, $t = 1$ implies $u = u'$. We may therefore assume that $s, t \notin \{0, 1\}$.

Suppose $u = v'$. Then either $u' = u$ or $\{u', u\} \in E$. In the latter case, $\varphi(u'), \varphi(u)$ are not linearly independent by assumption, which contradicts $t\varphi(u) = s\varphi(u')$. We may therefore assume that $u \neq v'$ and $v \neq u'$.

Since $s = t$ implies $u' = u$, we assume w.l.o.g. that $0 < s < t < 1$. It is easy to see that under these assumptions we have $\varphi(\{u, v\}) \cap \varphi(\{u', v'\}) \neq \emptyset$ (cf. Figure 5.1). As

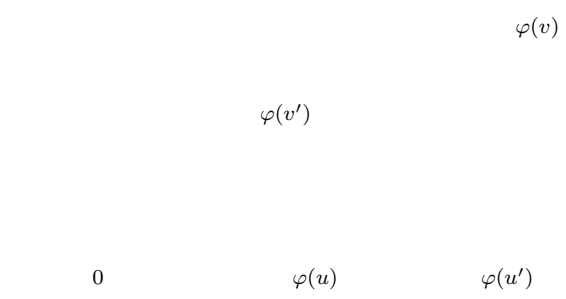

FIG. 5.1. *The segments* $\varphi(\{u, v\})$ *and* $\varphi(\{u', v'\})$ *intersect.*

φ is an embedding, we deduce from this $\{u,v\} \cap \{u',v'\} \neq \emptyset$, hence $u = u'$ or $v = v'$, which proves the claim.

The other cases to be treated are simpler and left to the reader. If $m > 2$, one can argue similarly. □

For positive integers a, b, p we define the graphs $G^p_{a,b} := K_a \otimes (K_b)^p$. Interpreting complete graphs as embedded via the standard simplices in Euclidean space and using the construction of the previous lemma, we get an embedding $\psi^p_{a,b}$ of $G^p_{a,b}$ in $\mathbb{R}^{abp}$. We denote the induced set of nodes in $\mathbb{R}^{abp}$ by R and the edge skeleton by T (omitting indices). The following property will be crucial.

LEMMA 5.7. *There are additive circuits of size polynomial in a, b, p depending uniformly on these parameters and performing the following tasks:*

1. *compute the map $\psi^p_{a,b} \colon [a] \times [b]^p \to R$ and its inverse,*
2. *decide in which set of the partition $\mathbb{R}^{abp} = (\mathbb{R}^{abp} \setminus T) \cup (T \setminus R) \cup R$ a given point in $\mathbb{R}^{abp}$ lies,*
3. *compute the end points of the (unique) edge segment of $G^p_{a,b}$ in which a given point $x \in T \setminus R$ lies.*

Proof. To compute $\psi^p_{a,b}$ is straightforward. For its inverse, as well as for parts (2) and (3), note that the relevant information can be inferred from the signs of the components x_i of a given point in $x \in \mathbb{R}^{abp}$. No multiplications, not even scalar ones, are needed to perform these tasks! □

5.2.3. Lower bound for reachability. We are going to show the $\mathrm{PAR}_{\mathrm{add}}$-hardness of $\mathrm{REACH}_{\mathrm{add}}$ and $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$. The next lemma tells us that it is sufficient to do this for $\mathrm{REACH}_{\mathrm{add}}$.

LEMMA 5.8. *The problem* $\mathrm{REACH}_{\mathrm{add}}$ *Turing reduces to* $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$.

Proof. Let $X \subseteq \mathbb{R}^n$ be given by an additive circuit $\mathcal{C}$ accepting X and suppose $s, t \in X$. Consider the following subset X' of $\mathbb{R}^{n+1}$:

$$X' := (X \times \{0\}) \cup (\{s\} \times [0,1]) \cup (\{t\} \times [0,1]) \cup (\mathbb{R}^n \times \{1\}).$$

There is an $\mathrm{FP}_{\mathrm{add}}$-machine, which takes as input a circuit $\mathcal{C}$ together with $s, t \in \mathbb{R}^n$ and outputs an additive circuit $\mathcal{C}''$ deciding membership to X'. It is easy to check that s and t are connected in X if and only if X' has the same number of connected components as X (and one less otherwise). The latter condition can be tested by querying a $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$-oracle twice, once with $\mathcal{C}$ and once with $\mathcal{C}''$. □

PROPOSITION 5.9. *The problem* $\mathrm{REACH}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-hard with respect to Turing reductions.*

Proof. By Theorem 4.1(3) it is sufficient to prove that $\mathrm{REACH}_{\mathrm{add}}$ is PSPACE-hard. Thus let $L \subseteq \{0,1\}^\infty$ be any language in PSPACE. Let M be a symmetric Turing machine deciding membership in L with polynomial space bound function $p(n)$ (cf. §5.2.1). For fixed input length n let H'_n denote the restriction of the configuration graph of M to the set of nodes $V_n := Q \times [p(n)] \times \Sigma^{p(n)}$. To an input $w \in \{0,1\}^n$ we assign the initial configuration $i(w) := (s, 1, (w_1, \ldots, w_n, \flat, \ldots, \flat))$. We may assume w.l.o.g. that there is exactly one accepting configuration $f := (t, 1, (\flat, \ldots, \flat))$. Of course, the cardinality of V_n is exponential in n. However, it is clear that the graph H'_n can be succinctly represented by Boolean circuits of size polynomial in n.

Note that if $(q, h, w), (q', h', w') \in V_n$ are two configurations adjacent in H'_n, then $|h - h'| \leq 1$ and the Hamming distance of w and w' is at most two. Let $\delta_2 = (p, ab, cd, q)$ be the transition between these configurations and w.l.o.g. $h' = h+1$. We introduce an additional node $(\delta, h, \tilde{w})$, where $\tilde{w}$ is obtained from w by changing

the hth entry to the one of w'. Thus the Hamming distances of both $w, \tilde{w}$ and $\tilde{w}, w'$ are at most one. We think of the node $(\delta, h, \tilde{w})$ as lying in between the nodes (q, h, w) and (q', h', w'). By this construction we obtain a *modified configuration graph* H_n on the set of nodes $V_n' := \tilde{Q} \times [p(n)] \times \Sigma^{p(n)}$ with the enlarged set of states $\tilde{Q} := Q \cup \Delta$. (Recall that Δ is the set of transitions of M.) If (q, h, w) and (q', h', w') are adjacent in H_n, then the Hamming distance of w and w' is at most one. Note that the graph H_n can also be succinctly represented by Boolean circuits of size polynomial in n.

By enumerating symbols we may assume that $\Sigma = [b]$ and $\tilde{Q} \times [p(n)] = [a(n)]$ with a polynomial function $a(n)$. From the above observations we conclude that the modified configuration graph H_n is a subgraph of the graph

$$G_n := G_{a(n),b}^{p(n)} := K_{a(n)} \otimes (K_b)^{p(n)}$$

defined in §5.2.2. We embed G_n in $\mathbb{R}^{a(n)p(n)b}$ with the construction of §5.2.2 using a map $\psi_n \colon \tilde{Q} \times [p(n)] \times \Sigma^{p(n)} \to R_n$ with induced set of nodes R_n. We denote the edge skeleton of this embedding by T_n, and denote by S_n the edge skeleton of the induced embedding of the subgraph H_n of G_n.

By Lemma 5.7, membership in T_n can be decided by a uniform family of additive circuits of size polynomial in n. It is now easy to see that also membership in S_n can be decided by such a family $(\mathcal{C}_n)$ of additive circuits. In fact, we first find out whether a given point $x \in \mathbb{R}^{a(n)p(n)b}$ lies in R_n or T_n. If x lies in $T_n \setminus R_n$, then we compute the end points y, z of the (unique) edge segment of G_n in which x lies. Furthermore, we compute the inverse images $\eta := \psi_n^{-1}(y)$ and $\zeta := \psi_n^{-1}(z)$. Due to Lemma 5.7 all this can be done by a uniform family of additive circuits of size polynomial in n. Note that x lies in S_n iff η and ζ are adjacent in the modified configuration graph H_n. Since the latter can be succinctly described by Boolean circuits of size polynomial in n we can test this in polynomial time.

Consider the map φ associating $w \in \{0,1\}^n$ to $(\mathcal{C}_n, \psi_n(i(w)), \psi_n(f))$. By construction, $w \in L$ iff the configurations $i(w)$ and f can be connected in the modified configuration graph H_n. This in turn is equivalent to the statement that the points $\psi_n(i(w))$ and $\psi_n(f)$ are connected by a path in the skeleton S_n defined by $\mathcal{C}_n$. Therefore, φ is a reduction from L to $\mathrm{REACH}_{\mathrm{add}}$.

In addition, it is obvious that the additive circuit $\mathcal{C}_n$ for S_n as well as the points $\psi_n(i(w))$ and $\psi_n(f)$ can be computed in polynomial time from w. This completes the proof of the proposition. □

REMARK 5.10. From the above proof it follows immediately that the problems $\mathrm{REACH}_{\mathrm{add}}$ and $\#^{\mathrm{cc}}\mathrm{CSAT}_{\mathrm{add}}$ restricted to closed input sets X remain complete for $\mathrm{PAR}_{\mathrm{add}}$ and $\mathrm{FPAR}_{\mathrm{add}}$, respectively.

5.2.4. Upper bound for reachability. In order to finish the proof of Theorem 5.3, it remains to show the following lemma.

LEMMA 5.11. *The problem* $\mathrm{REACH}_{\mathrm{add}}$ *is contained in* $\mathrm{PAR}_{\mathrm{add}}$ *and the problem* $\#^{\mathrm{cc}}\mathrm{CSAT}_{\mathrm{add}}$ *is contained* $\mathrm{FPAR}_{\mathrm{add}}$.

Proof. Let $\mathcal{C}$ be an additive circuit defining a set X. The computation of $\mathcal{C}$ can be unwound to a linear decision tree. We define a graph G, whose nodes consist of the accepting leaves of this tree, and whose edges join two different leaves μ and ν iff the corresponding leaf sets D_μ and D_ν touch each other, that is, $\overline{D_\mu} \cap D_\nu \neq \emptyset$ or $D_\mu \cap \overline{D_\nu} \neq \emptyset$.

Let $K_1, \ldots, K_t$ be the connected components of the graph G. It is easy to see that X has exactly t connected components, namely the sets of the form $\bigcup_{\nu \in K_i} D_\nu$

for $i \in [t]$. (Use that leaf sets are convex and thus connected.) Therefore, the number of connected components of X is equal to the number of connected components of the graph G.

Of course, the graph G may be exponentially large. However, it can be represented by a weaker variant of the succinct representation discussed before. The nodes of G can be encoded by a word in $\{0,1\}^\infty$ encoding the corresponding path in the tree (0 means branching to the left, 1 means branching to the right). For two such given nodes μ, ν, we can decide in $\mathrm{NP}_{\mathrm{add}}$ whether they are connected in G by guessing a point $x \in \mathbb{R}^n$ and checking whether $x \in \overline{D_\mu} \cap D_\nu$ or $x \in D_\mu \cap \overline{D_\nu}$. The latter can be done as follows: we can easily write down the linear functions computed along the path of μ, thus obtaining a description of D_μ by a system of linear inequalities. A system describing the closure $\overline{D}_\mu$ can be obtained from the one describing D_μ by relaxing the occuring inequalites $<$ to $\leq$.

As in the proof of Savage's theorem we can decide in $\mathrm{PAR}_{\mathrm{add}}$ whether two nodes of G are connected by a path as follows. Let the predicate $\mathrm{PATH}(\mu, \nu, i)$ express that the nodes μ and ν are connected by a path of length at most 2^i. Then we implement $\mathrm{PATH}(\mu, \nu, i)$ by the recursive algorithm

`for all nodes` ω `test whether` $\mathrm{PATH}(\mu, \omega, i-1)$ `and` $\mathrm{PATH}(\omega, \nu, i-1)$

using only polynomial space (compare [34, p. 149]). By applying this procedure for every pair of nodes of G we can compute the number of connected components of G and thus that of X in $\mathrm{FPAR}_{\mathrm{add}}$. □

REMARK 5.12. Let p be a prime. Then the problem of counting the number of connected components modulo p of a semi-linear set given by an additive circuit is also $\mathrm{FPAR}_{\mathrm{add}}$-complete with respect to Turing reductions. For showing this, we only have to observe that the proof of Lemma 5.8 immediately extends to counting mod p.

5.3. Euler characteristic and Betti numbers. The main results of this section are the completeness results for $\mathrm{EULER}_{\mathrm{add}}$ and $\mathrm{BETTI}_{\mathrm{add}}(k)$ treated in §5.3.4 and §5.3.5. The following subsections prepare for the proofs.

5.3.1. Cell complexes and homology. We recall some notions from algebraic topology [20, 33]. A *cell* of dimension k is a topological space homeomorphic to the open k-dimensional unit ball $\mathrm{int}(B^k) := \{x \in \mathbb{R}^k \mid x_1^2 + \cdots x_k^2 < 1\}$. The closed unit ball will be denoted by B^k. Its boundary ∂B^k is homeomorphic to the $(k-1)$-dimensional unit sphere.

Assume that a topological Hausdorff space X is decomposed into a finite, disjoint union of cells: $X = \cup_{i=1}^N F_i$. The *k-skeleton* X^k is then defined as the union of the cells of dimension at most k. The cell decomposition is called a *finite cell complex* iff each cell F_i has a characteristic map, that is, a continuous map $h_i \colon B^k \to X$ mapping the boundary ∂B^k to X^{k-1} and such that h_i induces a homeomorphism of $\mathrm{int}(B^k)$ with F_i. In this case, X is necessarily compact.

In the following, we assume that X is a compact, semi-linear subset of $\mathbb{R}^n$ decomposed into subsets F_i, $i \in [N]$, each described by a system

$$f_1 = a_1, \ldots, f_r = a_r,\ g_1 > d_1, \ldots, g_s > d_s,$$

where f_i, g_j are linear forms. Note that the F_i are bounded convex sets, which are open in their affine closure $\mathrm{aff}(F_i)$. In particular, each F_i is a cell and ∂F_i is homeomorphic to a sphere. It is easy to see that this cell decomposition of X is a finite cell complex

if the following *boundary condition* is satisfied:

$$\forall i, j \in [N] \quad F_i \cap \partial F_j \neq \emptyset \Longrightarrow F_i \subseteq \partial F_j. \tag{5.2}$$

This condition is equivalent to saying that the boundary ∂F_i of each cell is a union of cells. Such cell complexes will be called *semi-linear cell complexes* in the sequel.

The following fact is well known [20, 33]. Let X be decomposed as a semi-linear cell complex and denote by c_k the number of the k-cells of this decomposition. Then the Euler characteristic $\chi(X)$ can be computed as $\chi(X) = \sum_{k=0}^{n}(-1)^k c_k$.

We remark that the decomposition into leaf sets given by a ternary additive circuit may violate the boundary condition, which is a source of complications for our investigations. (For a definition of ternary circuits, see §5.3.3.) For instance, consider the triangle X decomposed into the vertices $a := (0,0), b := (2,0), c := (0,2)$ and the open segments segments joining a with b, a with c, b with c, and a with $(1,1)$. Then the boundary point $(1,1)$ of the open segment joining a with $(1,1)$ is not a vertex.

In order to define the cellular homology of a semi-linear cell complex, we first need to recall some facts about orientation.

Recall that an ordered basis of a finite dimensional real vector space defines an *orientation* of this space. Two ordered bases are said to have the same orientation iff the transformation matrix sending one basis to the other has positive determinant. By an orientation of an affine linear subspace $A \subseteq \mathbb{R}^n$ we understand an orientation of its associated linear space $L(A)$. An orientation of a convex subset F of $\mathbb{R}^n$ is defined as an orientation of its affine hull $\mathrm{aff}(F)$. It will be convenient to write $L(F) := L(\mathrm{aff}(F))$.

Let $A \subseteq \mathbb{R}^n$ be given as the zero set of linear polynomials $f_1, \ldots, f_{n-k}$ in this order. Extend the sequence $f_1 - f_1(0), \ldots, f_{n-k} - f_{n-k}(0)$ to a basis of the space of linear forms such that that the corresponding dual basis $(v_1, \ldots, v_n)$ is a positively oriented basis of $\mathbb{R}^n$. Then $(v_{n-k+1}, \ldots, v_n)$ is a basis of the linear space $L(A)$, which we define to be positively oriented. We will say that this is the orientation of A *induced by* $f_1, \ldots, f_{n-k}$. (Note that this is well defined.)

Let now A be the convex hull of a convex subset F of $\mathbb{R}^n$ and assume that H is a supporting hyperplane of F in A. That is, F lies on one side of H and that the closure of F meets H. Then an *orientation of F induces an orientation of H* as follows: let y be a vector pointing from H outward of F. Then we say that a basis $v_1, \ldots, v_{n-1}$ of $L(H)$ is positively oriented with respect to the induced orientation iff $y, v_1, \ldots, v_{n-1}$ is a positively oriented basis of $L(A)$. (This is again well defined.)

Let now $X = \cup_{i=1}^{N} F_i$ be a semi-linear cell complex and assume that all the cells F_i are oriented. Let Φ_k denote the set of k-cells and consider $F' \in \Phi_k$ and $F \in \Phi_{k+1}$. Assume that F' is contained in the closure of F. Then the affine hull of F' is a hyperplane in the affine hull of F supporting the convex set F. Therefore, the orientation of F induces an orientation on F' as explained above.

We define the *incidence number* $[F, F']$ of $F \in \Phi_{k+1}$ and $F' \in \Phi_k$ by

$$[F, F'] = \begin{cases} 0 & \text{iff } F' \text{ is not contained in the closure of } F, \\ 1 & \text{if the orientation } F \text{ induces on } F' \text{ is the same as that of } F', \\ -1 & \text{otherwise.} \end{cases}$$

For $k \geq 0$ the *incidence matrix* I_k is the matrix associated to $I_k \colon \Phi_{k+1} \times \Phi_k \to \mathbb{Z}$ by $I_k(F, F') = [F, F']$.

Let $\mathcal{C}_k$ be the $\mathbb{Q}$-vector space having Φ_k as a basis. The boundary map $\partial_k \colon \mathcal{C}_{k+1} \to \mathcal{C}_k$ is the $\mathbb{Q}$-linear map defined for $F \in \Phi_{k+1}$ by

$$\partial_k(F) = \sum_{F' \in \Phi_k} [F, F']\, F'.$$

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 $H_k := Z_k/B_k$. It is well known [20, 33] that H_k is isomorphic to the singular homology vector space $H_k(X;\mathbb{Q})$. Therefore, $b_k(X) := \dim H_k$ is the kth Betti number, which is independent of the cell decomposition and of the choice of orientations for its cells. We have $b_k(X) = \dim Z_k - \dim B_k = c_k - \rho_{k-1} - \rho_k$, where $\rho_k := \mathrm{rank}\,\partial_k$ and $c_k = |\Phi_k|$.

5.3.2. Reduction to the compact case. The technical result developed in this section will be needed in the upper bound proofs for the problems $\mathrm{EULER}_{\mathrm{add}}$ and $\mathrm{BETTI}_{\mathrm{add}}(k)$. The purpose is to show that the closedness assumption on X can be strengthened to compactness without loss of generality.

Let us first show that both closedness and compactness of a semi-linear set can be checked within the allowed resources, that is, in additive polynomial time with access to a $\#\mathsf{P}$-oracle. We write $\|x\|_\infty := \max_{i \le n} |x_i|$ for $x \in \mathbb{R}^n$.

LEMMA 5.13. *Both closedness and compactness of a set X given by an additive circuit can be decided in $\mathrm{P}_{\mathrm{add}}^{\#\mathsf{P}}$.*

Proof. The boundedness of $X \subseteq \mathbb{R}^n$ can be expressed as follows:

$$\exists R \in \mathbb{R}\ \forall x\ (x \in X \Longrightarrow \|x\|_\infty \le R).$$

Hence this property can be decided in Σ^2_{add}. The closedness of $X \subseteq \mathbb{R}^n$ can be expressed by:

$$\forall y\ \exists \epsilon\ \forall x\ (y \notin X, x \in X, \epsilon > 0 \Longrightarrow \|x - y\|_\infty \ge \epsilon).$$

Hence this property can be decided in Π^3_{add}. Now use Corollary 4.6. □

We recall a further notion from topology [20, 33]. A subspace A of a space X is called a *strong deformation retract* of X if there is a continous map $F \colon X \times [0,1] \to X$ such that $F(x,0) = x$, $F(x,1) \in A$, and $F(a,t) = a$ for all $x \in X$, $a \in A$ and $t \in [0,1]$. It is a well-known fact that, if A is a strong deformation retract of X, then the inclusion of A in X induces an isomorphism of the homology vector spaces $H_k(A;\mathbb{Q}) \simeq H_k(X;\mathbb{Q})$. In particular, the spaces A and X have the same Betti numbers.

LEMMA 5.14. *Let $X \subseteq \mathbb{R}^n$ be a closed semi-linear set given by an additive circuit $\mathcal{C}$. As usual, we denote by D_ν the corresponding leaf sets. Then:*

1. *We can compute from $\mathcal{C}$ in $\mathrm{FP}_{\mathrm{add}}$ a real number $R > 0$ such that*

$$\forall \nu\ \big(D_\nu \neq \emptyset \Longrightarrow D_\nu \cap [-R,R]^n \neq \emptyset\big). \tag{5.3}$$

2. *If R satisfies (5.3), then $X_R := X \cap [-R,R]^n$ is a strong deformation retract of X.*

Proof. 1. Each leaf set D_ν is defined by a set of sign conditions for the values computed by the circuit. On an input $y \in \mathbb{R}^n$ these values are of the form

$$z = \sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{k} b_i \alpha_i + c,$$

where $\alpha_1, \ldots, \alpha_k$ are the constants of $\mathcal{C}$ and the coefficients a_i, b_i, c are integers of bit-size at most $s := \mathrm{size}(\mathcal{C})$. If D_ν is not empty, Theorem 2.6 implies that there is a point $y \in D_\nu$ such that $y_i = \sum_{j=1}^k u_{ij}\alpha_j + w_i$, where the u_{ij}, w_i are rationals of bit-size at most $L := (sn)^c$, c being some universal constant. Hence $\max_i |y_i| \leq R$, where $R := 2^L\big(1 + \sum_{j=1}^k |\alpha_j|\big)$. Therefore, the bound R satisfies the condition (5.3). Moreover, it is clear that R can be computed in $\mathrm{FP}_{\mathrm{add}}$ from $\mathcal{C}$.

2. Assume w.l.o.g. that X is not compact. Consider the one-point compactification $\dot{X}$ of X, which is explicitly defined as follows. Let $S^n \subset \mathbb{R}^{n+1}$ be the n-dimensional sphere and $N = (0, 0, \ldots, 0, 1) \in S^n$ be its north pole. Projection from N yields a homeomorphism between $S^n - \{N\}$ and $\mathbb{R}^n \times \{-1\}$, and therefore a homeomorphism between $S^n - \{N\}$ and $\mathbb{R}^n$. The closure $\dot{X}$ of the image of X in S^n (which consists of attaching N to this image) is called a *one-point compactification* of X. The decomposition into leaf sets of X becomes a cell decomposition of $\dot{X}$, which can be turned into a finite cell complex by refinement. The claim is now a consequence of the following intuitive topological fact, whose formal proof is left to the reader. Let Y be a finite cell complex and p be a vertex of Y. Assume that U is an open neighborhood of p so small, that $\{p\}$ is the only cell of the complex contained in U. Then $Y \setminus U$ is a strong deformation retract of $Y \setminus \{p\}$. □

5.3.3. Universal cell decompositions. We adopt the notation $\mathcal{F}_{s,n}$ for the universal cell decomposition for the parameters s, n, introduced in §4.1.

LEMMA 5.15. *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 semi-linear cell complex.*

Proof. It is obvious that the boundary condition (5.2) is satisfied. □

Let $\mathcal{C}'$ be an additive circuit defining the semi-linear set $X \subseteq \mathbb{R}^n$. At the price of at most doubling the size of the circuit, we can transform $\mathcal{C}'$ into a *ternary additive circuit* $\mathcal{C}$, which branches according to the sign of intermediate results in a ternary way ($< 0, = 0, > 0$) instead of branching in a binary way according to $x \geq 0$ or $x < 0$. Each (non-empty) leaf set D_ν of $\mathcal{C}$ is described in the form

$$f_1 = a_1, \ldots, f_r = a_r,\ g_1 > d_1, \ldots, g_s > d_s,$$

where the $f_i - a_i$ and $g_j - d_j$ are the linear polynomials computed along the path leading to the leaf ν. Note that the linear forms f_i, g_j have integer coefficients of bit-size at most 2^s, where s is the size of the circuit $\mathcal{C}$. If the circuit uses only the constants $0, 1$, then a_i, d_j are also integers of bit-size at most 2^s. In the general case, however, a_i, d_j are real numbers. In the first case, each leaf set D_ν is a union of faces of $\mathcal{F}_{s,n}$. Thus, $\{F \in \mathcal{F}_{s,n} \mid F \subseteq X\}$ is a refinement of the decomposition of X into the leaf sets. By Lemma 5.15 this decomposition is a semi-linear cell complex if X is compact.

Let X be a compact finite union of cells of $\mathcal{F}_{s,n}$. To define (and compute) the cellular homology groups of X we need to fix orientations on the cells $F \in \mathcal{F}_{s,n}$. The cellular homology groups are independent of the chosen orientations, so we will make this choice in a convenient way as explained below.

By identifying a sequence $(f_1, \ldots, f_k)$ in $(\mathcal{H}_{s,n})^k$ with the sequence of coefficients of $f_1, \ldots, f_k$ (in a fixed order), and using the lexicographical ordering, we may consider $(\mathcal{H}_{s,n})^k$ as a totally ordered set. We can extend this order to the union $\mathcal{H}_{s,n}^\infty$ of the $(\mathcal{H}_{s,n})^k$, for $k \in \mathbb{N}$, by requiring that elements of $(\mathcal{H}_{s,n})^k$ are strictly smaller than elements of $(\mathcal{H}_{s,n})^{k'}$, for $k < k'$.

For $F \in \mathcal{F}_{s,n}$ let $(f_1, \ldots, f_{n-k})$ be the smallest sequence in $\mathcal{H}_{s,n}^\infty$ such that F is contained in the zero set of $f_1, \ldots, f_{n-k}$. Then $k = \dim F$. We define the orientation

of F as the orientation of $\mathrm{aff}(F)$ induced by this smallest sequence $(f_1, \ldots, f_{n-k})$ (cf. §5.3.1).

In the sequel, $\mathcal{F}$ shall denote the union of the $\mathcal{F}_{s,n}$, and $\mathcal{H}$ the union of the $\mathcal{H}_{s,n}$, over all $s, n \in \mathbb{N}$, respectively. Recall that we encode the faces $F \in \mathcal{F}$ by triples $(s, n, x) \in \mathbb{N}^2 \times \mathbb{Q}^n$ with a rational point $x \in F$ of bit-size at most $(sn)^c$. Let the *incidence function* $I\colon \mathcal{F} \times \mathcal{F} \to \{-1, 0, 1\}$ by defined by $I(F, F') := [F, F']$, if $F, F' \in \mathcal{F}_{s,n}$ for some s, n and $\dim F = \dim F' + 1$, and $I(F, F') = 0$ otherwise.

Lemma 5.16.

1. *The membership decision problem* $\{(F, x) \in \mathcal{F} \times \mathbb{R}^\infty \mid x \in F\}$ *is in* $\mathrm{PH}^0_{\mathrm{add}}$.
2. *The containement decision problem* $\{(F, f) \in \mathcal{F} \times \mathcal{H} \mid F \subseteq Z(f)\}$ *is in* PH.

Here $Z(f)$ *denotes the zero set of the polynomials of the sequence* f.

3. *The closure containement problem* $\{(F', F) \in \mathcal{F} \times \mathcal{F} \mid F' \subseteq \partial F\}$ *is in* PH.
4. *There is a function* $\mathcal{F} \to \mathbb{Q}^\infty$ *in* $\mathsf{FP}^{\mathsf{PH}}$ *mapping* F *to a positively oriented basis of the linear space* $L(F)$ *associated with the affine hull of* F.
5. *The incidence function* $I\colon \mathcal{F} \times \mathcal{F} \to \{-1, 0, 1\}$ *can be computed in* $\mathsf{FP}^{\mathsf{PH}}$.

Proof. 1. Let $F \in \mathcal{F}_{s,n}$ be given by the rational point $x_0 \in F$. We have

$$x \in F \iff \forall f \in \mathcal{H}_{s,n}\ \big(\mathrm{sgn} f(x) = \mathrm{sgn} f(x_0)\big).$$

This condition is expressible in $(\Pi^1_{\mathrm{add}})^0 \subseteq \mathrm{PH}^0_{\mathrm{add}}$, which shows the claim.

2. Replacing in the statement $\forall x\ (x \in F \Rightarrow x \in Z(f))$ the predicate "$x \in F$" according to (1), we obtain a $(\Pi^1_{\mathrm{add}})^0$-statement. Since F and f are discrete, the containement decision problem even belongs to the Boolean part of $(\Pi^1_{\mathrm{add}})^0$, and thus to Π^1 by [14] (cf. Remark 4.12).

3. Using (1), we can express "$x \in \overline{F}$" by a $(\Pi^2_{\mathrm{add}})^0$-statement. Hence $F' \subseteq \partial F$ can also be expressed by a $(\Pi^2_{\mathrm{add}})^0$-statement. Since F, F' are discrete, this statement is even in Π^2.

4. Using (2), we see that the following condition is in Π^2: $(f_1, \ldots, f_{n-k})$ is the smallest sequence in $\mathcal{H}^\infty_{s,n}$ such that F is contained in the zero set of $f_1, \ldots, f_{n-k}$. The assertion follows now by Remark 2.11(ii).

5. For given $F, F' \in \mathcal{F}_{s,n}$ we first check in PH whether $F' \subseteq \partial F$ and $\dim F' = \dim F - 1$ using parts (3) and (4). Then we compute positively oriented bases $u_1, \ldots, u_{k+1}$ of $L(F)$ and $v_1, \ldots, v_k$ of $L(F')$ in $\mathsf{FP}^{\mathsf{PH}}$ according to part (4). If F and F' are represented by the rational points x and x', respectively, then $y := x' - x$ is a vector in $L(F)$ pointing from $L(F')$ outside of F. The incidence number $[F, F']$ equals 1 iff the bases $y, v_1, \ldots, v_k$ and $u_1, \ldots, u_{k+1}$ have the same orientation, which can be checked in P. □

We extend now what we have discussed before to the case, when there are real constants.

Let $s, n, \ell \in \mathbb{N}$ and $\eta \in \mathbb{R}^\ell$. By intersecting the universal cell decomposition of $\mathbb{R}^{n+\ell}$ for the parameters $s, n + \ell$ with the hyperplane $H(\eta) := \{(x, y) \in \mathbb{R}^{n+\ell} \mid y = \eta\}$, we get a cell decomposition of $\mathbb{R}^n \times \{\eta\}$, which we identify with $\mathbb{R}^n$. More specifically, each face $F \in \mathcal{F}_{s+\ell,n}$ induces a face $F(\eta)$ of this decomposition, defined by $F(\eta) \times \{\eta\} := F \cap H(\eta)$, provided this intersection is non-empty. Note that each $F(\eta)$ is defined by putting sign conditions on all polynomials of the form $a_0 + \sum_{j=1}^{\ell} b_j \eta_j + \sum_{i=1}^{n} a_i X_i$, where $\sum_{i=0}^{n} |a_i| + \sum_{j=1}^{\ell} |b_j| \le 2^s$. We write $\mathcal{F}_{s,n}(\eta) := \{F(\eta) \mid F \in \mathcal{F}_{s,n}\}$ and call this the *universal cell decomposition* of $\mathbb{R}^n$ for the parameters s, n and vector of constants $\eta \in \mathbb{R}^\ell$. Moreover, we write $\mathcal{F}(\eta)$ for the union of the $\mathcal{F}_{s,n}(\eta)$ over all $s, n \in \mathbb{N}$.

Most of the results shown so far in this subsection extend to this more general notion of universal cell decompositions in a natural way. For instance, Lemma 5.15 extends immediately to $\mathcal{F}_{s,n}(\eta)$. A face $F(\eta) \in \mathcal{F}(\eta)$ is encoded by $F \in \mathcal{F}$, which is itself encoded by a small rational point.

LEMMA 5.17. *An analogue of Lemma 5.16 holds, where the complexity classes* $\mathrm{PH}^0_{\mathrm{add}}$, FP, *and* PH *have to be replaced by* $\mathrm{PH}_{\mathrm{add}}$, $\mathsf{FP/poly}$, *and* $\mathsf{PH/poly}$, *respectively.*

The proof is a straightforward extension of the proof of Lemma 5.16. For instance, for treating the closure containement problem $\{(F', F) \in \mathcal{F}(\eta) \times \mathcal{F}(\eta) \mid F' \subseteq \partial F\}$ one first shows that this problem is in $\mathrm{PH}_{\mathrm{add}}$. Then, since this is a discrete problem, one concludes that it is in the Boolean part of $\mathrm{PH}_{\mathrm{add}}$, and thus in $\mathsf{PH/poly}$ by [14] (cf. Remark 4.12).

5.3.4. Euler characteristic. The Euler characteristic $\chi(X)$ is a fundamental invariant of a topological space.

Let $\mathrm{EULER}_{\mathrm{add}}$ denote the following problem: given an additive circuit $\mathcal{C}$ defining a *closed* semi-linear set X, decide whether X is empty and if not, compute its Euler characteristic $\chi(X)$. Hence only circuits defining closed semi-linear sets are considered to be admissible inputs.

THEOREM 5.18. *The problem* $\mathrm{EULER}_{\mathrm{add}}$ *is* $\mathrm{FP}^{\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}}$*-complete with respect to Turing reductions.*

Proof. We first show that $\mathrm{EULER}_{\mathrm{add}}$ is $\#\mathrm{P}_{\mathrm{add}}$-hard. Note that the problem $\mathrm{CSAT}_{\mathrm{add}}$ introduced in §1.4 trivially reduces to $\mathrm{EULER}_{\mathrm{add}}$ by the definition of the latter problem. Hence $\mathrm{NP}_{\mathrm{add}} \subseteq \mathrm{P}^{\mathrm{EULER}_{\mathrm{add}}}_{\mathrm{add}}$. Therefore, by Theorem 5.1, we have $\mathrm{DIM}_{\mathrm{add}}(1) \in \mathrm{P}^{\mathrm{EULER}_{\mathrm{add}}}_{\mathrm{add}}$.

It is now easy to design a Turing reduction from $\#\mathrm{CSAT}_{\mathrm{add}}$ to $\mathrm{EULER}_{\mathrm{add}}$. On input an additive circuit $\mathcal{C}$, first decide whether X is finite using oracle calls to $\mathrm{DIM}_{\mathrm{add}}(1)$, and hence to $\mathrm{EULER}_{\mathrm{add}}$. If no, return ∞, otherwise return $\chi(X)$. Since $\#\mathrm{CSAT}_{\mathrm{add}}$ is $\#\mathrm{P}_{\mathrm{add}}$-complete by Theorem 3.6, this proves the hardness.

It remains to prove that $\mathrm{EULER}_{\mathrm{add}}$ is contained in $\mathrm{FP}^{\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}}$. By Lemma 5.14, we may restrict our discussion to additive circuits defining a compact semi-linear set $X \subseteq \mathbb{R}^n$. Assume that X is given by a ternary additive circuit $\mathcal{C}$ of size s using the real constants $\eta_1, \ldots, \eta_\ell$. Then each of the leaf sets of $\mathcal{C}$ is a union of faces in $\mathcal{F}_{s,n}(\eta)$. Hence the decomposition of X into the faces $F \in \mathcal{F}_{s,n}(\eta)$ contained in X is a semi-linear cell complex. If $c_k(\mathcal{C})$ denotes the number of k-cells of this cell complex, we have $\chi(X) = \sum_{k=0}^{n} (-1)^k c_k(\mathcal{C})$.

Lemma 5.16 and its extension Lemma 5.17 imply that on input $\mathcal{C}$ and $F \in \mathcal{F}_{s,n}(\eta)$, the property $F \in \Phi_k(\mathcal{C})$ can be tested in $\mathrm{DPH}_{\mathrm{add}}$. Therefore, $c_k(\mathcal{C})$ can be computed from $\mathcal{C}$ in $\mathrm{D}\#\cdot\mathrm{PH}_{\mathrm{add}}$, which is contained in $\mathrm{FP}^{\#\mathsf{P}}_{\mathrm{add}}$ by Corollary 4.6. This shows that $\mathrm{EULER}_{\mathrm{add}}$ belongs to $\mathrm{FP}^{\#\mathsf{P}}_{\mathrm{add}}$. □

5.3.5. Betti numbers. The kth Betti number $b_k(X)$ of a space X is defined as the dimension of the kth (singular) homology vector space $H_k(X; \mathbb{Q})$ ($k \in \mathbb{N}$). The Betti numbers modulo a prime p are defined by replacing the coefficient field $\mathbb{Q}$ by the finite field $\mathbb{F}_p$.

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. Recall that for $k = 0$ this is just the problem of counting the number of connected components. The problem of computing the kth Betti number modulo a prime p shall be denoted by $\mathrm{BETTI}_{\mathrm{add}}(k, \bmod p)$.

The goal of this section is the proof of the following result, extending Theorem 5.3.

THEOREM 5.19. *For any $k \in \mathbb{N}$ and any prime p, the problems* $\mathrm{BETTI}_{\mathrm{add}}(k)$ *and* $\mathrm{BETTI}_{\mathrm{add}}(k, \mathrm{mod}\, p)$ *are* $\mathrm{FPAR}_{\mathrm{add}}$*-complete with respect to Turing reductions.*

The next lemma provides the lower bound part of the proof of Theorem 5.19.

LEMMA 5.20. $\mathrm{BETTI}_{\mathrm{add}}(k)$ *and* $\mathrm{BETTI}_{\mathrm{add}}(k, \mathrm{mod}\, p)$ *are* $\mathrm{FPAR}_{\mathrm{add}}$*-hard with respect to Turing reductions, for any $k \in \mathbb{N}$ and any prime p.*

Proof. We will exhibit a Turing reduction from $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$ to $\mathrm{BETTI}_{\mathrm{add}}(k)$. Without loss of generality, we assume $k > 0$.

The *suspension* $S(X)$ of a topological space X is defined as the space obtained from the cylinder $X \times [0,1]$ over X by identifying the points in each of the sets $X \times \{0\}$ and $X \times \{1\}$. Essentially, this is a double cone with basis X. It is well known that, if $X \neq \emptyset$, the Betti numbers of $S(X)$ and X are related as follows (cf. [20, 33]):

$$b_{k+1}(S(X)) = \begin{cases} b_k(X) & \text{if } k > 0 \\ b_0(X) - 1 & \text{if } k = 0. \end{cases} \tag{5.4}$$

If $X \subseteq \mathbb{R}^n$ is given by an additive circuit, then we will use the following alternative definition for the suspension

$$S_1(X) := (X \times [0,1]) \cup (\mathbb{R}^n \times \{0\}) \cup (\mathbb{R}^n \times \{1\}),$$

which, if $X \neq \emptyset$, is homotopy equivalent to $S(X)$ and therefore has the same Betti numbers. Also,

$$b_k(S_1(\emptyset)) = \begin{cases} 0 & \text{if } k > 0 \\ 2 & \text{if } k = 0. \end{cases} \tag{5.5}$$

This alternative definition of suspension has the advantage that it is easy to transform an additive circuit describing X to one describing $S_1(X)$. Note that $S_1(X)$ is closed if X is closed. If we iterate this construction $k+1$ times starting with $X \subseteq \mathbb{R}^n$, we get a set $S_{k+1}(X) \subseteq \mathbb{R}^{n+k+1}$, which satisfies, by (5.4) and (5.5),

$$b_k(S_{k+1}(X)) = b_0(S_1(X)) - 1 = \begin{cases} 0 & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset. \end{cases}$$

This allows one to decide whether $X = \emptyset$ by one query to $\mathrm{BETTI}_{\mathrm{add}}(k)$. If $X = \emptyset$ then we return 0. Otherwise, note that, by (5.4), $b_0(X) = b_k(S_k(X)) + 1$ (since $k > 0$) and we may return $b_0(X)$ after another query to $\mathrm{BETTI}_{\mathrm{add}}(k)$. Strictly speaking, this is a reduction from the restriction of $\#^{\mathsf{cc}}\mathrm{CSAT}_{\mathrm{add}}$ to closed input sets X. However, this is sufficient by Remark 5.10.

The same reduction can be made for the Betti numbers modulo a prime. The hardness follows in this case by appealing to Remark 5.12. □

Before proving the upper bound, we make a short digression on space efficient linear algebra. It is well known [32] that the rank of an $N \times N$ integer matrix A, whose entries have bit-size at most L, can be computed by uniform Boolean circuits with depth $(\log L + \log N)^{O(1)}$ (and similary for matrices over $\mathbb{F}_p$). Using Borodin's result [6], this can be translated to a polylogarithmic space computation of the rank of A by a Turing machine.

Similarly as for graphs, we will understand by a *succinct representation* of an integer matrix $A = (a_{ij})$ a Boolean circuit B computing the matrix entry a_{ij} from the index pair (i, j) given in binary. From the above discussion we conclude the following lemma.

LEMMA 5.21. *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 bit-size of the entries of A.*

We finish now the proof of Theorem 5.19 by showing the following upper bound.

PROPOSITION 5.22. $\mathrm{BETTI}_{\mathrm{add}}(k)$ *and* $\mathrm{BETTI}_{\mathrm{add}}(k, \bmod p)$ *are both contained in* $\mathrm{FPAR}_{\mathrm{add}}$.

Proof. 1. We first prove that $\mathrm{BETTI}^0_{\mathrm{add}}(k)$ is in $\mathsf{FPSPACE}$. Let the closed semi-linear set $X \subseteq \mathbb{R}^n$ be given by an additive circuit $\mathcal{C}'$, whose only constants are 0 and 1. By Lemma 5.14, we may assume w.l.o.g. that X is compact. We transform $\mathcal{C}'$ into a ternary additive circuit $\mathcal{C}$ of size s as in §5.3.3. Consider the cell decomposition of X induced by $\mathcal{C}$. It is clear that each of its leaf sets is a union of faces in $\mathcal{F}_{s,n}$. Lemma 5.15 implies that the decomposition of X into the faces $F \in \mathcal{F}_{s,n}$ contained in X is a semi-linear cell complex.

We put $\Phi_k(\mathcal{C}) := \{F \in \mathcal{F}_{s,n} \mid \dim F = k, F \subseteq X\}$ and $c_k(\mathcal{C}) := |\Phi_k(\mathcal{C})|$ for $k \in \mathbb{N}$. Let $\rho_k(\mathcal{C})$ denote the rank of the incidence matrix $I_k(\mathcal{C})\colon \Phi_{k+1}(\mathcal{C}) \times \Phi_k(\mathcal{C}) \to \mathbb{Z}, (F, F') \mapsto [F, F']$. In §5.3.1 it was shown that the kth Betti number $b_k(X)$ of X can be expressed as $b_k(X) = c_k(\mathcal{C}) - \rho_{k-1}(\mathcal{C}) - \rho_k(\mathcal{C})$.

Lemma 5.16 implies that on input $\mathcal{C}$ and F, the property $F \in \Phi_k(\mathcal{C})$ can be tested in PH. Therefore, $c_k(\mathcal{C})$ can be computed from $\mathcal{C}$ in $\# \cdot \mathsf{PH} \subseteq \mathsf{FPSPACE}$.

Thus it remains to show that for each $k \in \mathbb{N}$, the function $\mathcal{C} \mapsto \rho_k(\mathcal{C})$ can be computed in $\mathsf{FPSPACE}$. It follows from Lemma 5.16(5) that $I_k(\mathcal{C})(F, F')$ can be computed from $\mathcal{C}, F, F'$ in $\mathsf{FPSPACE}$. Hence, by Borodin's result [6], a succinct representation B of the incidence matrix $I_k(\mathcal{C})$ having depth polynomial in s can be computed from $\mathcal{C}$ in $\mathsf{FPSPACE}$. By Lemma 5.21, we can compute the rank $\rho_k(\mathcal{C})$ from B by a Turing machine in space polynomial in s. Altogether, we get a computation of $\rho_k(\mathcal{C})$ in $\mathsf{FPSPACE}$.

2. We now prove that $\mathrm{BETTI}_{\mathrm{add}}(k)$ belongs to $\mathrm{FPAR}_{\mathrm{add}}$. Assume that the compact semi-linear set $X \subseteq \mathbb{R}^n$ is given by a ternary additive circuit $\mathcal{C}$ of size s using the real constants $\eta_1, \ldots, \eta_\ell$. Then each of the leaf sets of $\mathcal{C}$ is a union of faces in $\mathcal{F}_{s,n}(\eta)$. Hence the decomposition of X into the faces $F \in \mathcal{F}_{s,n}(\eta)$ contained in X is a semi-linear cell complex. The rest of the argument is based on Lemma 5.17 and similar as before.

3. The case of positive characteristic can be settled similarly. □

COROLLARY 5.23. $\mathrm{BETTI}_{\mathrm{add}}(k)$ *Turing reduces to* $\mathrm{EULER}_{\mathrm{add}}$ *for some* $k \in \mathbb{N}$ *iff* $\mathsf{PSPACE} \subseteq \mathsf{P}^{\#\mathsf{P}}/\mathsf{poly}$*. It does so with a constant-free reduction iff* $\mathsf{PSPACE} \subseteq \mathsf{P}^{\#\mathsf{P}}$.

Proof. This follows from Theorem 5.18, Theorem 5.19, and Corollary 4.11. □

5.3.6. Some completeness results in the Turing model. The results of §4 and §5.1–§5.3 can be combined to show completeness results for natural geometric problems in the discrete setting.

An additive circuit $\mathcal{C}$ whose only constants are 0 and 1 can be encoded in $\{0,1\}^\infty$. Thus, one may consider discrete versions $\mathrm{EULER}^0_{\mathrm{add}}$ and $\mathrm{BETTI}^0_{\mathrm{add}}(k)$, which are defined as $\mathrm{EULER}_{\mathrm{add}}$ and $\mathrm{BETTI}_{\mathrm{add}}(k)$ respectively, but restricted to constant-free circuits. Note that, since $\mathrm{BP}((\Pi^3_{\mathrm{add}})^0) = \Pi^3$ (cf. [14]), a corresponding version of Lemma 5.13 holds, i.e., closedness and compactness of a set given by a constant-free additive circuit can be decided in $\mathsf{P}^{\#\mathsf{P}}$.

For these discrete problems the following results hold.

COROLLARY 5.24. *The problems* $\mathrm{EULER}^0_{\mathrm{add}}$ *and* $\mathrm{BETTI}^0_{\mathrm{add}}(k)$, $k \in \mathbb{N}$, *are complete with respect to Turing reductions in* $\mathsf{FP}^{\#\mathsf{P}}$ *and* $\mathsf{FPSPACE}$*, respectively. A similar*

statement holds for the computation of Betti numbers modulo a prime.

Proof. This could be shown by checking in detail the proofs of Theorem 5.18 and Theorem 5.19. More elegantly, we can derive Corollary 5.24 from general principles using the concept of Boolean parts. We then only need to check that the reductions in the proofs of Theorem 3.6, Theorem 5.18, and Theorem 5.19 are constant-free. For instance, $\mathrm{EULER}^0_{\mathrm{add}} \in \mathrm{BP}((\mathrm{FP}^0_{\mathrm{add}})^{\#\mathrm{P}^0_{\mathrm{add}}}) = \mathsf{FP}^{\#\mathsf{P}}$ according to Remark 4.12. For the hardness note that

$$(\mathrm{FP}^0_{\mathrm{add}})^{\#\mathrm{P}^0_{\mathrm{add}}} = (\mathrm{FP}^0_{\mathrm{add}})^{\#\mathrm{CSAT}^0_{\mathrm{add}}} = (\mathrm{FP}^0_{\mathrm{add}})^{\mathrm{EULER}^0_{\mathrm{add}}}.$$

By taking Boolean parts und using Remark 4.12, we conclude $\mathsf{FP}^{\#\mathsf{P}} = \mathsf{FP}^{\mathrm{EULER}^0_{\mathrm{add}}}$.

The statement about $\mathrm{BETTI}^0_{\mathrm{add}}(k)$ is proved similarly. □

6. Summary. To facilitate the orientation of the reader, we have summarized the results of this paper in Figure 6.1. There, an arrow denotes an inclusion, problems in square brackets are Turing-complete for the class at their left, and problems in curly brackets are many-one-complete for that class. The problems appearing in the figure are defined in the list below. We note that the completeness of SAT, $\mathrm{CSAT}_{\mathrm{add}}$, $\mathrm{TSP}_{\mathbb{R}}$, $\mathrm{PM}_{\mathbb{R}}$, QBF, and DTRAO was already known.

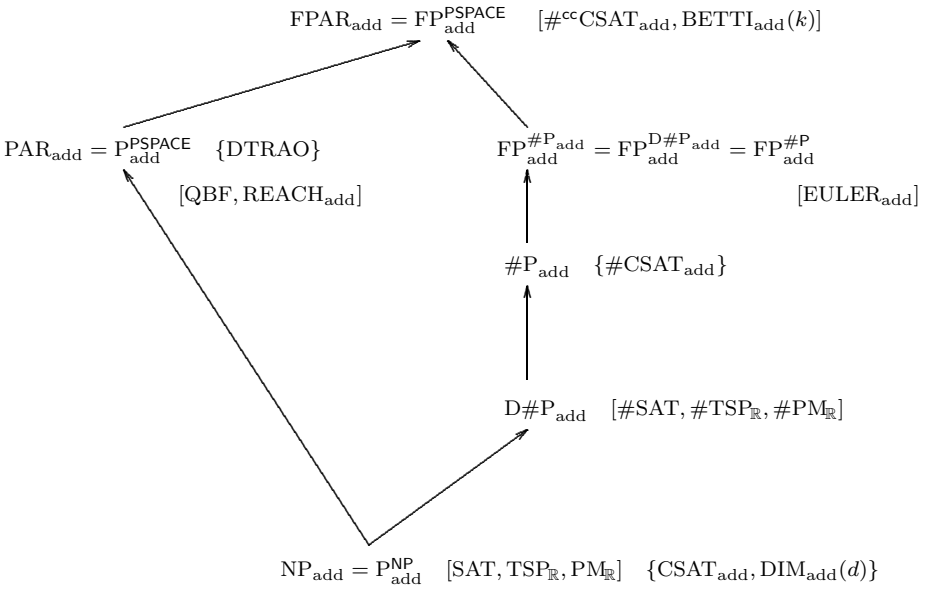

FIG. 6.1. *Survey of main results.*

SAT (*Satisfiability*) Given a propositional formula φ, decide whether there is an assignment of Boolean values for the variables satisfying φ.

#SAT (*Counting Satisfiability*) Given φ as in SAT, count the number of satisfying assignments.

QBF (*Quantified Boolean Formulas*) Given a quantified Boolean formula, decide whether it is a tautology.

DTRAO (*Digital Theory of the Reals with Addition and Order*) Given a sentence in the theory of the reals with addition and order, all of whose variables satisfy a constraint of the form $x = 0 \vee x = 1$, decide whether it is a tautology.

$\mathrm{TSP}_{\mathbb{R}}$ (*Traveling Salesman*) Given a complete graph G with real weights on the edges and $w \in \mathbb{R}$, decide whether there is a Hamilton circuit in G with weight at most w. (The weight of a subgraph is the sum of the weights of its edges.)

$\#\mathrm{TSP}_{\mathbb{R}}$ (*Counting Traveling Salesman*) Given G and $w \in \mathbb{R}$ as in $\mathrm{TSP}_{\mathbb{R}}$, count the number of Hamilton circuits in G with weight at most w.

$\mathrm{PM}_{\mathbb{R}}$ (*Weighted Perfect Matching*) Given a bipartite graph G with real weights on the edges and $w \in \mathbb{R}$, decide whether there is a perfect matching in G with weight at most w.

$\#\mathrm{PM}_{\mathbb{R}}$ (*Counting Weighted Perfect Matchings*) Given G and $w \in \mathbb{R}$ as in $\mathrm{PM}_{\mathbb{R}}$, count the number of perfect matchings in G with weight at most w.

$\mathrm{CSAT}_{\mathrm{add}}$ (*Circuit Satisfiability*) Decide whether the semi-linear set given by an additive circuit is non-empty.

$\mathrm{DIM}_{\mathrm{add}}(d)$ (*Dimension*) Given an additive circuit and $d \in \mathbb{N}$, decide whether the dimension of the semi-linear set defined by the circuit is at least d.

$\mathrm{REACH}_{\mathrm{add}}$ (*Reachability*) Given an additive circuit defining a semi-linear set X and two points $s, t \in X$, decide whether s and t are in the same connected component of X.

$\#\mathrm{CSAT}_{\mathrm{add}}$ (*Point Counting*) Given an additive circuit defining a semi-linear set X, compute the number of points in X.

$\#^{\mathrm{cc}}\mathrm{CSAT}_{\mathrm{add}}$ (*Counting Connected Components*) Given an additive circuit defining a semi-linear set X, compute the number of connected components of X.

$\mathrm{EULER}_{\mathrm{add}}$ (*Euler Characteristic*) Given an additive circuit defining a closed semi-linear set X, decide whether X is empty and if not, compute its Euler characteristic.

$\mathrm{BETTI}_{\mathrm{add}}(k)$ (*Betti Numbers*) Given an additive circuit defining a closed semi-linear set X, compute the kth Betti number of X.

7. Open questions. We present some selected open problems.

PROBLEM 7.1. 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 7.2. What is the complexity to deciding connectedness of a semi-linear set given by an additive circuit?

PROBLEM 7.3. In this paper we proved that computing the torsion-free part of the homology of semi-linear sets is $\mathrm{FPAR}_{\mathrm{add}}$-complete. Is the complexity of computing the torsion part of this homology also $\mathrm{FPAR}_{\mathrm{add}}$-complete?

REFERENCES

[1] J. BALCÁZAR, A. LOZANO, AND J. TORÁN, *The complexity of algorithmic problem on succinct instances*, in Computer Science: Research and Applications, Plenum Press, New York, 1992, pp. 351–377.

[2] J. L. BALCÁZAR, J. D. AZ, AND J. GABARRÓ, *Structural Complexity I*, Springer Verlag, 1988.

[3] M. BEN-OR, *Lower bounds for algebraic computation trees*, in Proc. 15th ACM STOC, Boston, 1983, pp. 80–86.

[4] L. BLUM, F. CUCKER, M. SHUB, AND S. SMALE, *Complexity and Real Computation*, Springer, 1998.

[5] L. BLUM, M. SHUB, AND S. SMALE, *On a theory of computation and complexity over the real numbers*, Bull. Amer. Math. Soc., 21 (1989), pp. 1–46.

[6] A. BORODIN, *On relating time and space to size and depth*, SIAM J. Comp., 6 (1977), pp. 733–744.

[7] P. BÜRGISSER, *Completeness and Reduction in Algebraic Complexity Theory*, vol. 7 of Algorithms and Computation in Mathematics, Springer Verlag, 2000.

[8] ———, *Cook's versus Valiant's hypothesis*, Theoret. Comp. Sci., 235 (2000), pp. 71–88.

[9] ———, *Lower bounds and real algebraic geometry*, in Algorithmic and Quantitative Real Algebraic Geometry, S. Basu and L. Gonzales-Vega, eds., vol. 60 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS, 2003.

[10] P. Bürgisser, M. Clausen, and M. Shokrollahi, *Algebraic Complexity Theory*, vol. 315 of Grundlehren der mathematischen Wissenschaften, Springer Verlag, 1997.
[11] P. Bürgisser and F. Cucker, *Counting complexity classes for numeric computations II: Algebraic and semialgebraic sets*. In preparation.
[12] F. Cucker and D. Grigoriev, *On the power of real Turing machines over binary inputs*, SIAM J. Comp., 26 (1997), pp. 243–254.
[13] F. Cucker, M. Karpinski, P. Koiran, T. Lickteig, and K. Werther, *On real Turing machines that toss coins*, in Proc. 27th ACM STOC, Las Vegas, 1995, pp. 335–342.
[14] F. Cucker and P. Koiran, *Computing over the reals with addition and order: Higher complexity classes*, J. Compl., 11 (1995), pp. 358–376.
[15] F. Cucker and M. Matamala, *On digital nondeterminism*, Mathematical Systems Theory, 29 (1996), pp. 635–647.
[16] H. Fournier and P. Koiran, *Are lower bounds easier over the reals?*, in Proc. 30th ACM STOC, 1998, pp. 507–513.
[17] ———, *Lower bounds are not easier over the reals: Inside* PH, in Proc. ICALP 2000, LNCS 1853, 2000, pp. 832–843.
[18] H. Galperin and A. Wigderson, *Succinct representation of graphs*, Information and Control, 56 (1983), pp. 183–198.
[19] J. v. z. Gathen, *Parallel arithmetic computations: a survey*, in Proc. 12th Symp. Math. Found. Comput. Sci., Bratislava, no. 233 in LNCS, 1986, pp. 93–112.
[20] A. Hatcher, *Algebraic topology*, Cambridge University Press, Cambridge, 2002.
[21] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[22] L. Khachijan, *A polynomial algorithm in linear programming*, Dokl. Akad. Nauk SSSR, 244 (1979), pp. 1093–1096. (In Russian.) English translation in *Soviet Math. Dokl.* 20:191-194,1979.
[23] K.-I. Ko, *Complexity of Real Functions*, Birkhäuser, 1991.
[24] P. Koiran, *Computing over the reals with addition and order*, Theoret. Comp. Sci., 133 (1994), pp. 35–47.
[25] ———, *A weak version of the Blum, Shub & Smale model*, J. Comp. Syst. Sci., 54 (1997), pp. 177–189.
[26] H. Lewis and C. Papadimitriou, *Symmetric space-bounded computation*, Theoret. Comp. Sci., 19 (1982), pp. 161–187.
[27] K. Meer, *Counting problems over the reals*, Theoret. Comp. Sci., 242 (2000), pp. 41–58.
[28] S. Meiser, *Point location in arrangements of hyperplanes*, Information and Computation, 106 (1993), pp. 286–303.
[29] F. Meyer auf der Heide, *A polynomial linear search algorithm for the n-dimensional knapsack problem*, J. ACM, 31 (1984), pp. 668–676.
[30] ———, *Fast algorithms for n-dimensional restrictions of hard problems*, J. ACM, 35 (1988), pp. 740–747.
[31] 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 (1989), pp. 435–437.
[32] K. Mulmuley, *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Combinatorica, 7 (1987), pp. 101–104.
[33] J. R. Munkres, *Elements of algebraic topology*, Addison-Wesley Publishing Company, Menlo Park, CA, 1984.
[34] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.
[35] J. Reif, *Complexity of the mover's problem and generalizations*, in Proc. 20th FOCS, 1979, pp. 421–427.
[36] ———, *Complexity of the generalized mover's problem*, in Planning, Geometry and Complexity of Robot Motion, J. Schwartz, M. Sharir, and J. Hopcroft, eds., Ablex Publishing Corporation, 1987, pp. 267–281.
[37] E. Tardos, *A strongly polynomial algorithm to solve combinatorial linear programs*, Oper. Res., 34 (1986), pp. 250–256.
[38] S. Toda, PP *is as hard as the polynomial-time hierarchy*, SIAM J. Comp., 21 (1991), pp. 865–877.
[39] S. Toda and O. Watanabe, *Polynomial time 1-Turing reductions from #*PH *to #*P, Theoret. Comp. Sci., 100 (1992), pp. 205–221.
[40] L. Valiant, *Completeness classes in algebra*, in Proc. 11th ACM STOC, 1979, pp. 249–261.
[41] ———, *The complexity of computing the permanent*, Theoret. Comp. Sci., 8 (1979), pp. 189–201.
[42] ———, *The complexity of enumeration and reliability problems*, SIAM J. Comp., 8 (1979),

pp. 410–421.
[43] ———, *Reducibility by algebraic projections*, in Logic and Algorithmic: an International Symposium held in honor of Ernst Specker, vol. 30, Monogr. No. 30 de l'Enseign. Math., 1982, pp. 365–380.
[44] A. Yao, *Algebraic decision trees and Euler characteristic*, in Proc. 33rd FOCS, 1992.
[45] ———, *Decision tree complexity and Betti numbers*, in Proc. 26th ACM STOC, 1994.