The Complexity of Semilinear Problems in Succinct Representation

(Extended Abstract)*

Peter Bürgisser**[1], Felipe Cucker***[2], and Paulin Jacobé de Naurois

[1] Dept. of Mathematics, University of Paderborn, D-33095 Paderborn, Germany
pbuerg@upb.de

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

[3] LORIA, 615 rue du Jardin Botanique, BP 101, 54602 Villers-lès-Nancy Cedex, Nancy, France
Paulin.De-Naurois@loria.fr

Abstract. We prove completeness results for twenty-three problems in semilinear geometry. These results involve semilinear sets given by additive circuits as input data. If arbitrary real constants are allowed in the circuit, the completeness results are for the Blum-Shub-Smale additive model of computation. If, in contrast, the circuit is constant-free, then the completeness results are for the Turing model of computation. One such result, the $\mathsf{P}^{\mathsf{NP}[\log]}$-completeness of deciding Zariski irreducibility, exhibits for the first time a problem with a geometric nature complete in this class.

1 Introduction and Main Results

A subset $S \subseteq \mathbb{R}^n$ is *semilinear* if it is a Boolean combination of closed half-spaces $\{x \in \mathbb{R}^n \mid a_1x_1 + \ldots + a_nx_n \leq b\}$. That is, S is derived from closed half-spaces by taking a finite number of unions, intersections, and complements.

The geometry of semilinear sets and its algorithmics has been a subject of interest for a long time not the least because of its close relationship with linear programming and its applications. This relationship is at the heart of many algorithmic results on both semilinear geometry and linear programming. It is also a good starting point to motivate the results in this paper.

Consider the feasibility problem for linear programming. That is, the problem of deciding whether a system of linear equalities and inequalities has a solution. A celebrated result by Khachijan [8] states that if the coefficients of these equalities and inequalities are integers then this problem can be solved in polynomial time

---

* A full version of this paper can be obtained at http://www-math.upb.de/agpb

** Partially supported by DFG grant BU 1371 and Paderborn Institute for Scientific Computation (PaSCo).

*** Partially supported by City University SRG grant 7001558.

in the Turing machine model; that is, it belongs to the class P. If the coefficients are not integers but arbitrary real numbers, the Turing machine model is no longer appropriate. Instead, we analyze this version of the problem using the machine model over the real numbers introduced by Blum, Shub and Smale (the BSS model in the following). While it is not difficult to show that the linear programming feasibility problem over $\mathbb{R}$ is in $\mathrm{NP}_{\mathbb{R}} \cap \mathrm{coNP}_{\mathbb{R}}$ (this is merely Farkas' Lemma), or even that it can be solved in average polynomial time, its membership to $\mathrm{P}_{\mathbb{R}}$ (i.e., its solvability in deterministic polynomial time in the BSS model) remains an open problem. This membership problem has even been proposed by Smale as one of the mathematical problems for the 21st century [17].

A situation intermediate between the two above is the one in which the inequalities $a_1x_1 + \ldots + a_nx_n \leq b$ have integer coefficients a_i and real right hand side b. In this case, the appropriate model of computation is the *additive model*. This is a restriction of the BSS model over $\mathbb{R}$ where multiplications and divisions are excluded from the capabilities of the machine. Only additions, subtractions and comparisons may be performed. The rephrasing of a well known result by Tardos [18] shows that the feasibility problem for a system of linear inequalities of the above mixed type is solvable in $\mathrm{P}_{\mathrm{add}}$.[1]

Equalities and inequalities of the mixed type we just described are not as rare as they may appear at a first glance. They naturally occur in the defining equations of semilinear sets given in *succinct representation*. Here, a semilinear set is given by an additive decision circuit (a more precise development follows in Section 2): a point $x \in \mathbb{R}^n$ is in the set if and only if the circuit returns 1 with input x. Since additive circuits are natural input data for additive machines one may wonder about the complexity of the feasibility problem $\mathrm{CS{\scriptsize AT}}_{\mathrm{add}}$ for semilinear sets in succinct representation. This problem consists of deciding whether the semilinear set S given by an additive circuit is nonempty. As it turns out, this problem is $\mathrm{NP}_{\mathrm{add}}$-complete [2]. This is in contrast with the result by Tardos mentioned above and is explained by the fact that an additive circuit of size $\mathcal{O}(n)$ can describe a semilinear set defined with $\mathcal{O}(2^n)$ linear inequalities.

The completeness result for $\mathrm{CS{\scriptsize AT}}_{\mathrm{add}}$ is not an isolated fact. It was recently shown [3] that several other problems for semilinear sets in succinct representation are complete in some complexity class. Notably, to decide whether the dimension of such a set is at least a given number is also $\mathrm{NP}_{\mathrm{add}}$-complete, to compute its Euler characteristic is $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$-complete, and to compute any of its Betti numbers is $\mathrm{FPAR}_{\mathrm{add}}$-complete.

One of the goals of this paper is to further expand the catalogue of complete problems in semilinear geometry. We will show completeness for twenty three problems in this domain. These results, together with the previous results mentioned above, draw an accurate landscape of the difficulty of different problems in semilinear geometry providing, at the same time, examples of natural

---

[1] The reader may have noticed that we use the subscript "add" for complexity classes in the additive model, the subscript "$\mathbb{R}$" for those in the unrestricted BSS model, and no subscript at all for those in the Turing model. In addition, to emphasize the latter, we use sanserif fonts.

complete problems for many of the complexity classes defined in the additive model.

A final remark is relevant. If an additive circuit has no constant gates (other than those with associated constant 0 or 1) it is said to be *constant-free*. Such a circuit can be described by means of a binary string and thus be taken as input by ordinary Turing machines. In this way, all problems considered in this paper have a discrete version fitting the classical complexity setting.

By checking our proofs one can see that all our completeness results hold for these discrete versions with respect to the corresponding discrete complexity classes.

We next briefly describe our main results. The precise definition of some concepts (e.g., Zariski topology) will be given later on this paper. The following list should give, however, an idea of the results we obtain. We consider the following problems related to topological properties of semilinear sets:

EADH$_{\text{add}}$ (*Euclidean Adherence*) Given a decision circuit $\mathscr{C}$ with n input gates and a point $x \in \mathbb{R}^n$, decide whether x belongs to the Euclidean closure of the semilinear set $S_{\mathscr{C}} \subseteq \mathbb{R}^n$ described by $\mathscr{C}$.

ECLOSED$_{\text{add}}$(*Euclidean Closed*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is closed under the Euclidean topology.

EDENSE$_{\text{add}}$(*Euclidean Denseness*) Given a decision circuit $\mathscr{C}$ with n input gates, decide whether $S_{\mathscr{C}}$ is dense in $\mathbb{R}^n$.

UNBOUNDED$_{\text{add}}$ (*Unboundedness*) Given a decision circuit $\mathscr{C}$ with n input gates, decide whether $S_{\mathscr{C}}$ is unbounded in $\mathbb{R}^n$.

COMPACT$_{\text{add}}$ (*Compactness*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is compact.

ISOLATED$_{\text{add}}$ (*Isolatedness*) Given a decision circuit $\mathscr{C}$ with n input gates and a point $x \in \mathbb{R}^n$, decide whether x is isolated in $S_{\mathscr{C}}$.

EXISTISO$_{\text{add}}$ (*Existence of Isolated Points*) Given a decision circuit $\mathscr{C}$ with n input gates, decide whether there exists $x \in \mathbb{R}^n$ isolated in $S_{\mathscr{C}}$.

#ISO$_{\text{add}}$ (*Counting Isolated Points*) Given a decision circuit $\mathscr{C}$, count the number of isolated points in $S_{\mathscr{C}}$.

LOCDIM$_{\text{add}}$ (*Local Dimension*) Given a decision circuit $\mathscr{C}$, a point $x \in S_{\mathscr{C}}$ and an integer $d \in \mathbb{N}$, decide whether $\dim_x S_{\mathscr{C}} \geq d$.

LOCCONT$_{\text{add}}$ (*Local Continuity*) Given an additive circuit $\mathscr{C}$ with n input gates and a point $x \in \mathbb{R}^n$, decide whether the function $F_{\mathscr{C}}$ computed by $\mathscr{C}$ is continuous at x (for the Euclidean topology).

CONT$_{\text{add}}$ (*Continuity*) Given an additive circuit $\mathscr{C}$, decide whether $F_{\mathscr{C}}$ is continuous (for the Euclidean topology).

SURJ$_{\text{add}}$ (*Surjectivity*) Given an additive circuit $\mathscr{C}$, decide whether $F_{\mathscr{C}}$ is surjective.

#DISC$_{\text{add}}$ (*Counting Discontinuities*) Given an additive circuit $\mathscr{C}$, count the number of points in $\mathbb{R}^n$ where $F_{\mathscr{C}}$ is not continuous for the Euclidean topology.

REACH$_{\text{add}}$ (*Reachability*) Given a decision circuit $\mathscr{C}$ with n input gates, and two points s and t in $\mathbb{R}^n$, decide whether s and t belong to the same connected component of $\mathcal{S}_{\mathscr{C}}$.

CONNECTED$_{\text{add}}$ (*Connectedness*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is connected.

TORSION$_{\text{add}}$ (*Torsion*) Given a decision circuit $\mathscr{C}$, decide whether the homology of $S_{\mathscr{C}}$ is torsion free.

ZADH_{add} (*Zariski Adherence*) Given a decision circuit $\mathscr{C}$ with n input gates and a point $x \in \mathbb{R}^n$, decide whether x belongs to the Zariski closure of $S_{\mathscr{C}}$.

$\text{ZCLOSED}_{\text{add}}$(*Zariski Closed*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is closed under the Zariski topology.

$\text{ZDENSE}_{\text{add}}$(*Zariski Denseness*) Given a decision circuit $\mathscr{C}$ with n input gates, decide whether $S_{\mathscr{C}}$ is Zariski dense in $\mathbb{R}^n$.

IRR_{add}(*Zariski Irreducibility*) Given a decision circuit $\mathscr{C}$, decide whether the Zariski closure of $S_{\mathscr{C}}$ is affine.

$\#\text{IRR}_{\text{add}}$ (*Counting Irreducible Components*) Given a decision circuit $\mathscr{C}$, count the number of irreducible components of $S_{\mathscr{C}}$.

$\#\text{IRR}^{(d)}_{\text{add}}$ (*Counting Irreducible Components of Fixed Dimension*) Given a decision circuit $\mathscr{C}$, count the number of irreducible components of $S_{\mathscr{C}}$ of dimension d.

$\#\text{IRR}^{[c]}_{\text{add}}$ (*Counting Irreducible Components of Fixed Codimension*) Given a decision circuit $\mathscr{C}$, count the number of irreducible components of $S_{\mathscr{C}}$ of codimension c.

$\#\text{IRR}^{\{N\}}_{\text{add}}$ (*Counting Irreducible Components in Fixed Ambient Space*) Given a decision circuit $\mathscr{C}$ with a fixed number N of input gates, count the number of irreducible components of $S_{\mathscr{C}}$.

Our main results can be summarized in the following table. Here (T) means that the hardness is for Turing reductions. In what follows, unless specified otherwise, completeness will always mean completeness with respect to many-one reductions.

Problems	**Complete in**	**Discrete version complete in**
EADH_{add}, ZADH_{add}	NP_{add}	NP
$\text{ECLOSED}_{\text{add}}$, $\text{ZCLOSED}_{\text{add}}$	coNP_{add}	coNP
$\text{EDENSE}_{\text{add}}$	coNP_{add}	coNP
$\text{ZDENSE}_{\text{add}}$	NP_{add}	NP
$\text{UNBOUNDED}_{\text{add}}$	NP_{add}	NP
$\text{COMPACT}_{\text{add}}$	coNP_{add}	coNP
$\text{ISOLATED}_{\text{add}}$	coNP_{add}	coNP
$\text{LOCDIM}_{\text{add}}$	NP_{add}	NP
$\text{LOCCONT}_{\text{add}}$, CONT_{add}	coNP_{add}	coNP
IRR_{add}	$\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$	$\mathsf{P}^{\mathsf{NP}[\log]}$
$\text{EXISTISO}_{\text{add}}$	Σ^2_{add}	$\Sigma_2\mathsf{P}$
SURJ_{add}	Π^2_{add}	$\Pi_2\mathsf{P}$
$\#\text{ISO}_{\text{add}}$, $\#\text{DISC}_{\text{add}}$	$\text{FP}_{\text{add}}^{\#\text{P}_{\text{add}}}$ (T)	$\mathsf{FP}^{\#\mathsf{P}}$ (T)
$\#\text{IRR}_{\text{add}}$, $\#\text{IRR}^{(d)}_{\text{add}}$, $\#\text{IRR}^{[c]}_{\text{add}}$, $\#\text{IRR}^{\{N\}}_{\text{add}}$	$\text{FP}_{\text{add}}^{\#\text{P}_{\text{add}}}$ (T)	$\mathsf{FP}^{\#\mathsf{P}}$ (T)
$\text{REACH}_{\text{add}}$, $\text{CONNECTED}_{\text{add}}$	PAR_{add} (T)	PSPACE

We remark that the Zariski topology and irreducible components are natural concepts studied in algebraic geometry [16]. In particular, we show that the problem to test irreducibility of a semilinear set given by a constant-free decision circuit is complete for the class $\mathsf{P}^{\mathsf{NP}[\log]}$. The latter class was first studied by Papadimitriou and Zachos [14] and consists of the decision problems that can be solved in polynomial time by $\mathcal{O}(\log n)$ queries to some NP language. Equivalently, $\mathsf{P}^{\mathsf{NP}[\log]}$ can also be characterized as the set of languages in P^{NP} whose queries are *non adaptive*, cf. [13, Th. 17.7]. This means that the input to any query does not depend on the oracle answer to previous queries, but only on the input of the machine. Several natural complete problems for $\mathsf{P}^{\mathsf{NP}[\log]}$ are known, see for instance [10, 7].

For the problem $\text{TORSION}_{\text{add}}$ we prove PAR_{add}-hardness (with respect to Turing reductions) and membership in EXP_{add} (PSPACE-hardness and membership in EXP for its discrete version). This advances towards determining the complexity of $\text{TORSION}_{\text{add}}$, a question left open in [3, §7]. Also, the PAR_{add}-completeness of $\text{CONNECTED}_{\text{add}}$ closes a question left open therein.

2 Preliminaries

We next review the notions which will be central in this paper, fixing notations at the same time. A basic reference (since this paper is an extension of it) is [3].

(1) The Euclidean norm in $\mathbb{R}^n$ induces a topology, called *Euclidean*, in $\mathbb{R}^n$. We will denote the closure of a subset $S \subseteq \mathbb{R}^n$ with respect to the Euclidean topology by $\overline{S}$. Following [16], we define another, coarser, topology in $\mathbb{R}^n$, hereby restricting us to semilinear sets.

Definition 1. *We call a semilinear set $S \subseteq \mathbb{R}^n$* Zariski closed *if it is a finite union of affine subspaces of $\mathbb{R}^n$. The* Zariski closure *of a semilinear set $V \subseteq \mathbb{R}^n$, denoted by $\overline{V}^Z$, is the smallest Zariski-closed semilinear subset of $\mathbb{R}^n$ containing V.*

The use of the words "closed" or "closure" is appropriate: the semilinear Zariski-closed sets satisfy the axioms of the closed sets of a topology on $\mathbb{R}^n$.

We will use the sign functions $\mathsf{sg} : \mathbb{R} \to \{-1, 0, 1\}$, $\mathsf{pos} : \mathbb{R} \to \{0, 1\}$ defined by $\mathsf{sg}(x) = \mathsf{pos}(x) = 1$ if $x > 0$, $\mathsf{sg}(0) = 0, \mathsf{pos}(0) = 1$, and $\mathsf{sg}(x) = -1, \mathsf{pos}(x) = 0$ if $x < 0$. We extend these functions to $\mathbb{R}^n$ componentwise. A *quadrant* of $\mathbb{R}^n$ is an open subset of $\mathbb{R}^n$ of the form $\{x \in \mathbb{R}^n \mid \mathsf{sg}(x) = \sigma\}$ for some $\sigma \in \{-1, 1\}^n$.

(2) We next recall a few facts concerning additive circuits. Such circuits are defined in many places [2, 3, 9]. An *additive circuit* is a directed acyclic graph whose nodes are of one of the following types: input, output, constant, addition, substraction, and selection. The first four types of node have an obvious semantics; selection nodes have four inputs v, a, b, c and return a if $v > 0$, b if $v = 0$ and c otherwise.

An additive circuit $\mathscr{C}$ with n input nodes and m output nodes computes a function $F_{\mathscr{C}} : \mathbb{R}^n \to \mathbb{R}^m$. A *decision circuit* $\mathscr{C}$ is an additive circuit with exactly one output node that is preceded by a selection node with $a, b, c \in \{0, 1\}$. Such a circuit computes a function $F_{\mathscr{C}} : \mathbb{R}^n \to \{0, 1\}$ and decides the semilinear set $S_{\mathscr{C}} := \{x \in \mathbb{R}^n \mid F_{\mathscr{C}}(x_1, \dots, x_n) = 1\}$. We say that $S_{\mathscr{C}}$ is given in *succinct representation.*

Definition 2. *Let $\mathscr{C}$ be a decision circuit with r selection gates and n input gates. A* path *γ of $\mathscr{C}$ is an element in $\{-1, 0, 1\}^r$. We say that $x \in \mathbb{R}^n$* follows *a path γ of $\mathscr{C}$ if, on input x and for all j, the result of the test performed at the j-th selection gate is γ_j (i.e., $\gamma_j = -1$ if the tested value v satisfies $v < 0$, $\gamma_j = 0$ if $v = 0$, and $\gamma_j = 1$ if $v > 0$). The* leaf set *of a path γ is defined as*

$$D_\gamma = \{x \in \mathbb{R}^n \mid \text{ input } x \text{ follows the path } \gamma \text{ of } \mathscr{C}\}.$$

A path γ is accepting *if and only if we have $F_{\mathscr{C}}(x) = 1$ for one (and hence for all) $x \in D_\gamma$. We denote by $\mathcal{A}_{\mathscr{C}}$ the set of accepting paths of the circuit $\mathscr{C}$.*

(3) We finally recall some notions of computation and complexity. In this paper we use additive machines (i.e., BSS machines over $\mathbb{R}$ which do not multiply or divide) as described in [2, Ch. 18] or in [9]. For these machines, versions of the usual complexity classes are defined yielding the classes $\mathrm{P}_{\mathrm{add}}$, $\mathrm{NP}_{\mathrm{add}}$, $\#\mathrm{P}_{\mathrm{add}}$, $\mathrm{PAR}_{\mathrm{add}}$, $\mathrm{EXP}_{\mathrm{add}}$, and $\mathrm{FP}_{\mathrm{add}}$ (note that the additive version of polynomial space requires instead polynomial parallel time). An overview of these classes and their properties can be found in [2, Ch. 18] and [3].

We already defined the problem $\mathrm{CSAT}_{\mathrm{add}}$ and observed that it is $\mathrm{NP}_{\mathrm{add}}$-complete. The following two problems are also $\mathrm{NP}_{\mathrm{add}}$-complete:

$\mathrm{CBS}_{\mathrm{add}}$ (*Circuit Boolean Satisfiability*) Given a decision circuit $\mathscr{C}$ with n input gates, decide whether there exists $x \in \{0,1\}^n$ such that $\mathscr{C}(x) = 1$.

$\mathrm{DIM}_{\mathrm{add}}$ (*Dimension*) Given a decision circuit $\mathscr{C}$ with n input gates and $k \in \mathbb{N}$, decide whether the dimension of $S_{\mathscr{C}}$ is greater than or equal to k.

For $\mathrm{DIM}_{\mathrm{add}}$ this follows easily from [3, Theorem 5.1] (there k is assumed to be fixed, but the proof carries over easily). Note that $\mathrm{CBS}_{\mathrm{add}}$ deals with a digital form of nondeterminism since it requires the circuit to be satisfied by a point in $\{0,1\}^n$.

The $\mathrm{NP}_{\mathrm{add}}$-completeness of $\mathrm{CBS}_{\mathrm{add}}$ allows us to use a problem with a discrete flavor to prove completeness results in the additive setting. More generally, a series of results starting in [6], continued in [3], and relying on Meyer auf der Heide [11], allow us to use standard discrete problems as basis for reductions yielding Turing-hardness results in the additive setting.

We finish these preliminaries with a lemma gathering several facts which will be used later on in many proofs.

Lemma 1. *Given a decision circuit $\mathscr{C}$, two paths γ, γ' of $\mathscr{C}$, and a point $x \in \mathbb{R}^n$, the following tasks can be performed by an additive machine in time polynomial in the size of $\mathscr{C}$:*

(i) *Decide whether D_γ is nonempty.*
(ii) *Decide whether $x \in \overline{D_\gamma}$, or decide whether $x \in \overline{D_\gamma}^Z$.*
(iii) *Compute* $\dim D_\gamma$.
(iv) *Decide whether $\overline{D_\gamma}^Z \subseteq \overline{D_{\gamma'}}^Z$.*

3 Some Proofs

In this section we give some proofs to convey an idea of our techniques.

3.1 Basic Topology

Proposition 1. *The problem* $\text{ZD}\textsc{ense}_{\text{add}}$ *is* NP_{add}*-complete.*

Proof. Note that $\overline{S_{\mathscr{C}}}^Z = \bigcup_{\gamma\in\mathcal{A}_{\mathscr{C}}} \overline{D_\gamma}^Z$. Therefore, $\overline{S}_{\mathscr{C}}^Z = \mathbb{R}^n$ if and only if there exists $\gamma \in \mathcal{A}_{\mathscr{C}}$ such that D_γ is Zariski dense in $\mathbb{R}^n$. Since $\overline{D_\gamma}^Z$ is the affine hull of D_γ (if $D_\gamma \neq \emptyset$), we see that D_γ is Zariski dense in $\mathbb{R}^n$ if and only if $\dim D_\gamma = n$. Hence, S is Zariski dense in $\mathbb{R}^n$ if and only if $\dim S = n$. The membership to NP_{add} now follows from the fact that $\text{D}\textsc{im}_{\text{add}}$ is in NP_{add}.

For proving the hardness, we reduce CBS_{add} to $\text{ZD}\textsc{ense}_{\text{add}}$. Assume $\mathscr{C}$ is a decision circuit with n input gates. Consider a circuit $\mathscr{C}'$ computing the function

$$G_{\mathscr{C}} \colon \mathbb{R}^n \to \{0,1\},\ x \mapsto F_{\mathscr{C}}(\mathsf{pos}(x)). \tag{1}$$

The mapping $\mathscr{C} \mapsto (\mathscr{C}', 0)$ reduces CBS_{add} to $\text{ZD}\textsc{ense}_{\text{add}}$. Indeed, if $S_{\mathscr{C}} \cap \{0,1\}^n = \emptyset$ then $S_{\mathscr{C}'} = \emptyset$ as well and hence $0 \notin \overline{S_{\mathscr{C}'}}$. On the other hand, if $S_{\mathscr{C}} \cap \{0,1\}^n \neq \emptyset$ then $S_{\mathscr{C}'}$ contains at least one quadrant and hence $0 \in \overline{S_{\mathscr{C}'}}$. □

The following result is proved with similar arguments.

Proposition 2. *The problem* $\text{E}\textsc{xist}\text{I}\textsc{so}_{\text{add}}$ *is* Σ^2_{add}*-complete.*

3.2 Zariski Irreducibility

Irreducibility is a natural concept in algebraic geometry [16]. For semilinear sets this notion can be defined as follows.

Definition 3. *A semilinear set* $S \subseteq \mathbb{R}^n$ *is* Zariski-irreducible *if its Zariski closure is an affine space. The Zariski closure of a semilinear set* $S \subseteq \mathbb{R}^n$ *is a non-redundant finite union of affine subspaces* $A_1, \dots, A_s$ *of* $\mathbb{R}^n$*. We call* $A_1, \dots, A_s$ *the* irreducible components *of* $\overline{S}^Z$ *and call the sets* $S \cap A_i$ *the* irreducible components *of* S*.*

We extend the definition of $\mathsf{P}^{\mathsf{NP}[\log]}$ to the additive setting in the obvious way thus obtaining the class $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$. Again, it is not difficult to show that this class can also be characterized as the set of decision problems solvable in additive polynomial time with non adaptive queries to NP_{add}.

The main result of this section is the following.

Theorem 1. *The problem* $\text{I}\textsc{rr}_{\text{add}}$ *is* $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$*-complete.*

We first prove the upper bound.

Lemma 2. *The problem* $\text{I}\textsc{rr}_{\text{add}}$ *is in* $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$*.*

Proof. Consider the following algorithm:

input $\mathscr{C}$ with n input gates
for $k = -1, \dots, n$ (independently) do
  (i) check whether $\dim S_{\mathscr{C}} \geq k$
  (ii) check whether $\forall \gamma, \gamma' \in \mathcal{A}_{\mathscr{C}}$ $(\dim D_{\gamma'} = k \Rightarrow \overline{D_\gamma}^Z \subseteq \overline{D_{\gamma'}}^Z)$
let $d = \max\{k : \text{ (i) holds }\}$
if (ii) holds for $k = d$ then ACCEPT else REJECT

This algorithm decides whether $S_{\mathscr{C}}$ is Zariski irreducible. Indeed, the dimension d of $S_{\mathscr{C}}$ is computed, and the query (ii) for $k = d$ checks whether for all leaf sets $D_{\gamma'}$ of dimension d we have $\overline{S_{\mathscr{C}}}^Z = \overline{D_{\gamma'}}^Z$. This holds if and only if $S_{\mathscr{C}}$ is Zariski irreducible.

Since DIM_{add} is known to be in NP_{add} [3], (i) is a query to a problem in NP_{add}. By Lemma 1, (ii) is a query to a problem in coNP_{add}. Since the queries are nonadaptive and the algorithm runs in polynomial time, the set IRR_{add} is in $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$. □

Lemma 3. (i) *Let $S_1 \subseteq \mathbb{R}^n$ and $S_2 \subseteq \mathbb{R}^m$ be two non-empty semilinear sets. Then, $S_1 \times S_2 \subseteq \mathbb{R}^{n+m}$ is irreducible if and only if both S_1 and S_2 are irreducible.*

(ii) *A nonempty union of reducible semilinear sets is reducible.* □

We turn now to the proof of the lower bound in Theorem 1.

Lemma 4. *The problem* IRR_{add} *is* $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$*-hard under many-one reductions.*

PROOF. Assume L is a problem in $\text{P}_{\text{add}}^{\text{NP}_{\text{add}}[\log]}$. Then we may assume that L is decided by a polynomial time additive machine asking non adaptively a polynomial number of queries to the NP_{add}-complete problem $\text{ZDENSE}_{\text{add}}$. Hence, there exists a polynomial p and, for all $n \in \mathbb{N}$, a polynomial size circuit $\mathscr{C}^n$ with $n + p(n)$ input gates and a family of polynomial size circuits $\mathscr{C}_1^n, \dots, \mathscr{C}_{p(n)}^n$ with n input gates, such that, for $x \in \mathbb{R}^n$, x is in L if and only if $F_{\mathscr{C}^n}(x, s) = 1$, where $s = (s_1, \dots, s_{p(n)})$ denotes the *sequence of oracle answers* for the input x, that is $s_i = 1$ if the output of $\mathscr{C}_i^n$ on input x is in $\text{ZDENSE}_{\text{add}}$ and $s_i = 0$ otherwise. Thus the circuits $\mathscr{C}_i^n$ compute the inputs to the oracle queries and $\mathscr{C}^n$ performs the final computation deciding the membership of x to L, given the sequence s of oracle answers.

The output $\mathscr{E}_i^n$ of $\mathscr{C}_i^n$ on input x is an input to $\text{ZDENSE}_{\text{add}}$. Thus $\mathscr{E}_i^n$ is a (description of a) decision circuit defining a semilinear set, which we denote by $S_i \subseteq \mathbb{R}^{r(n)}$. (Without loss of generality, we may assume that all these sets lie in a Euclidean space of the same dimension $r(n) > 1$ and that all the circuits $\mathscr{E}_i^n$ use the same number of selection gates $q(n) > 1$.) We denote by $\mathcal{A}_i$ the set of accepting paths of $\mathscr{E}_i^n$. Moreover, for $\gamma \in \mathcal{A}_i$, we denote by $D_{\gamma i} \subseteq \mathcal{S}_i$ the corresponding leaf set, and write $\partial D_{\gamma i}$ for its Euclidean boundary.

The reduction (1) from the proof of Proposition 1 that reduces CBS_{add} to $\text{ZDENSE}_{\text{add}}$ produces either a Zariski dense or an empty set. Moreover, the leaf sets produced by this reduction are, up to boundary points, quadrants of $\mathbb{R}^{r(n)}$.

Taking this into account, we may therefore assume without loss of generality that S_i is either empty or Zariski dense in $\mathbb{R}^{r(n)}$, for all $x \in \mathbb{R}^n$ and all i. Moreover, we may assume that (recall $r(n) > 1$)

$$S_i \neq \emptyset \Longrightarrow \bigcup_{\gamma \in \mathcal{A}_i} \partial D_{\gamma i} \text{ is reducible.} \tag{2}$$

Our goal is to reduce L to Irr_{add}. Thus we have to compute from $x \in \mathbb{R}^n$, in polynomial time, a decision circuit defining a semilinear set Ω such that $x \in L$ iff Ω is irreducible. We will consider $x \in \mathbb{R}^n$ as fixed and suppress it notationally. To simplify notation, we will write $p := p(n), q := q(n), r := r(n)$ for fixed $x \in \mathbb{R}^n$.

The set Ω will be a set of tuples (u, y, a) in the Euclidean space $\Pi := \mathbb{R}^q \times (\mathbb{R}^r)^p \times \mathbb{R}^p$. To convey an idea of the intended meaning, we call $u \in \mathbb{R}^q$ *selection gate vector*, $y = (y_1, \ldots, y_p) \in (\mathbb{R}^r)^p$ *oracle vector*, and $a \in \mathbb{R}^p$ *oracle answer vector*. A selection gate vector u induces a discrete vector $\gamma := \mathsf{sg}(u) \in \{-1, 0, 1\}^q$, which describes a possible path of one of the circuits $\mathcal{E}_i^n$. An oracle answer vector a induces a bit vector $\alpha := \mathsf{pos}(a) \in \{0, 1\}^p$, which describes a possible sequence of oracle answers. The set Ω will be a finite union of *product sets* of the form $U \times Y_1 \times \cdots \times Y_p \times A \subseteq \Pi$, where $U \subseteq \mathbb{R}^q$, $Y_i \subseteq \mathbb{R}^r$, and $A \subseteq \mathbb{R}^p$ are semilinear sets. Note that, by Lemma 3, a nonempty product set is irreducible iff all U, Y_i, A are irreducible and nonempty.

Let z be a fixed point in $\mathbb{R}^r$ (for instance the origin). Recall that $s \in \{0, 1\}^p$ denotes the sequence of oracle answers for the fixed input x. We define the subsets $T_i := S_i \cup \{z\} \subseteq \mathbb{R}^r$, for which we make the following important observation:

$$\begin{aligned} s_i = 1 &\Longleftrightarrow \overline{S_i}^Z = \mathbb{R}^r \Longleftrightarrow \overline{T_i}^Z = \mathbb{R}^r, \\ s_i = 0 &\Longleftrightarrow \quad S_i = \emptyset \quad \Longleftrightarrow \overline{T_i}^Z = \{z\}. \end{aligned} \tag{3}$$

We define the set $\Omega \subseteq \Pi$ as the one accepted by the following algorithm:

input $(u, y, a) \in \mathbb{R}^q \times (\mathbb{R}^r)^p \times \mathbb{R}^p$
compute $\gamma := \mathsf{sg}(u) \in \{-1, 0, 1\}^q$, $\alpha := \mathsf{pos}(a) \in \{0, 1\}^p$
(I) case $(\forall i\ y_i \in T_i) \ \wedge\ (\exists i\ a_i = 0)$ ACCEPT
(II) case $(F_{\mathscr{C}^n}(x, \alpha) = 1) \ \wedge\ (\forall i\ y_i \in T_i) \ \wedge\ \exists j\ \big(\alpha_j = 0 \ \wedge\ \gamma \in \mathcal{A}_j \ \wedge\ y_j \in \partial D_{\gamma j}\big)$ ACCEPT
(III) case $(F_{\mathscr{C}^n}(x, \alpha) = 1) \ \wedge\ \forall i\ \big((\alpha_i = 0 \implies y_i = z) \ \wedge\ (\alpha_i = 1 \implies y_i \in S_i)\big)$ ACCEPT
else REJECT.

It is easy to see that an additive circuit formalizing the above algorithm can be computed from the given $x \in \mathbb{R}^n$ in polynomial time by an additive machine. (Use that a description of the circuits $\mathscr{C}^n, \mathscr{C}_i^n$ can be computed from n by an additive machine in polynomial time.)

To prove the lemma, it is sufficient to show the following assertion:

$$x \in L \Longleftrightarrow \Omega \text{ is irreducible.} \tag{4}$$

In order to show this we are going to analyze the set Ω. We define

$$\Omega_{\text{I}} = \{(u, y, a) \in \Pi \mid \ (u, y, a) \text{ satisfies Case (I)}\}$$

and similarly Ω_{II} and Ω_{III}. Note that Ω_{II} is not the set of (u,y,a) accepted by the step (II) of the algorithm. We have $\Omega = \Omega_{\mathrm{I}} \cup \Omega_{\mathrm{II}} \cup \Omega_{\mathrm{III}}$, but this union is not necessarily disjoint. It is obvious that Ω_{I} is reducible.

We introduce some more notation needed for analyzing the above algorithm. Consider the following subset

$$\mathcal{Y} := \{\alpha \in \{0,1\}^p \mid F_{\mathscr{C}^n}(x,\alpha) = 1\}$$

of possible oracle answer sequences leading to acceptance. Note that $s \in \mathcal{Y}$ iff $x \in L$. Moreover, define for $\alpha \in \mathcal{Y}$ the following set of indices

$$J(\alpha) := \{j \mid \alpha_j = 0 \ \wedge \ s_j = 1\}$$

and for $j \in J(\alpha)$ let $\Omega^j_{\mathrm{II}}(\alpha)$ denote the set of $(u,y,a) \in \Pi$ that satisfy the condition of Case (II) with the α and j specified. Similarly, we define $\Omega_{\mathrm{III}}(\alpha)$. We have

$$\Omega = \Omega_{\mathrm{I}} \quad \cup \bigcup_{\alpha\in\mathcal{Y}, j\in J(\alpha)} \big(\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)\big). \tag{5}$$

The following claim settles one direction of (4).

Claim A. If $x \in L$, then Ω is irreducible.

In order to prove this claim, note that $\Omega_{\mathrm{III}}(s) = \mathbb{R}^q \times F_1 \times \cdots \times F_p \times \mathsf{pos}^{-1}(s)$, where we have put $F_i := S_i$ if $s_i = 1$ and $F_i := \{z\}$ otherwise. This implies that

$$\overline{\Omega_{\mathrm{III}}(s)}^Z = \mathbb{R}^q \times \overline{T_1}^Z \times \cdots \times \overline{T_p}^Z \times \mathbb{R}^p =: \Theta,$$

since $\overline{\mathsf{pos}^{-1}(s)}^Z = \mathbb{R}^p$. The product set Θ is irreducible by Lemma 3(i) and (3). It is clear that $\Omega_{\mathrm{I}} \cup \Omega_{\mathrm{II}} \subseteq \Theta$. Moreover, we claim that $\Omega_{\mathrm{III}}(\alpha) \subseteq \Theta$ for all $\alpha \in \mathcal{Y}$. Indeed, assume $(u,y,a) \in \Omega_{\mathrm{III}}(\alpha)$. If we had $s_i = 0$ and $\alpha_i = 1$ for some i, then we would have $y_i \in S_i$, which contradicts the fact that $S_i = \emptyset$ due to $s_i = 0$. This shows that $(u,y,a) \in \Theta$.

Altogether, using (5), we have shown that $\Omega \subseteq \Theta$. Hence $\overline{\Omega}^Z = \Theta$, which finishes the proof of Claim A.

Claim B. For $\alpha \in \mathcal{Y} \setminus \{s\}$, $j \in J(\alpha)$, the set $\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)$ is reducible.

Claim B implies the other direction of the assertion (4). Indeed, assume $x \notin L$. Then $s \notin \mathcal{Y}$ and according to (5), Ω is a union of reducible sets and thus reducible.

It remains to prove Claim B. Let $\pi_j : \Pi \to \mathbb{R}^r, (u,y,a) \to y_j$ be the projection onto the jth factor. In order to show that a subset $\Omega' \subseteq \Pi$ is reducible, it is sufficient to prove that $\pi_j(\Omega')$ is reducible, since irreducibility is preserved by linear maps. Hence it is enough to show that $\pi_j\big(\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)\big)$ is reducible. Taking into account (2) and the fact that $j \in J(\alpha)$ implies $S_j \neq \emptyset$, it suffices to prove that

$$\bigcup_{\gamma\in\mathcal{A}_j} \partial D_{\gamma j} \subseteq \pi_j\big(\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)\big) \subseteq \{z\} \cup \bigcup_{\gamma\in\mathcal{A}_j} \partial D_{\gamma j}.$$

The second inclusion is clear since $j \in J(\alpha)$ and thus $\alpha_j = 0$.

For the first inclusion, assume $y_j \in \partial D_{\gamma j}$ for some $\gamma \in \mathcal{A}_j$. Choose $a \in \mathbb{R}^p$ and $u \in \mathbb{R}^q$ such that $\mathsf{pos}(a) = \alpha$ and $\mathsf{sg}(u) = \gamma$. Then $(u, z, \dots, z, y_j, z, \dots, z, a) \in \Omega^j_{\mathrm{II}}(\alpha)$, where the y_j is at the jth position. Hence $y_j \in \pi_j\big(\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)\big)$. This finishes the proof of Claim B and completes the proof of the lemma. □

3.3 Problems of Connectivity

The proof of the following result is inspired by a similar result for graphs in [4].

Theorem 2. *The problem* $\textsc{Connected}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-complete under Turing reductions. The same holds when restricted to problems in* $\mathbb{R}^3$*.*

In [3] it was shown that, for all $k \in \mathbb{N}$, the problem to compute the kth Betti number of the semilinear set given by an additive circuit is $\mathrm{FPAR}_{\mathrm{add}}$-complete and the question was raised whether this holds also for the problem of computing the torsion subgroup of the homology group $H_k(X;\mathbb{Z})$. We give a partial answer to this question by showing that this problem is in fact $\mathrm{FPAR}_{\mathrm{add}}$-hard. Hereby we focus on the problem $\textsc{Torsion}_{\mathrm{add}}$ of deciding whether the torsion subgroups $T_k(S_{\mathscr{C}})$ of a semilinear set $S_{\mathscr{C}}$ given by a circuit vanish for all k, that is, whether all the homology groups $H_k(S_{\mathscr{C}};\mathbb{Z})$ are free abelian groups. The question of the corresponding upper bound remains open, but at least we show that the problem is in $\mathrm{EXP}_{\mathrm{add}}$.

Theorem 3. *The problem* $\textsc{Torsion}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-hard under Turing reductions and belongs to* $\mathrm{EXP}_{\mathrm{add}}$*.*

For the lower bound proof, we start with the reduction in the proof of Theorem 2, which reduces any language L in PSPACE to $\textsc{Connected}_{\mathrm{add}}$ by mapping a bit string x to a decision circuit describing a semilinear set $S'_n \subseteq \mathbb{R}^3$ such that S'_n is connected iff $x \in L$. Then we extend this construction by modifying the space $S'_n \times [0,1]$ roughly by building in a Moebius strip and identifying the boundary lines of the resulting space.

4 Open Problems

Let the semilinear set $S_{\mathscr{C}}$ be given by a constant-free decision circuit $\mathscr{C}$. We remark that the problem to test simple connectivity of $S_{\mathscr{C}}$ is undecidable. This follows by reducing to it the group triviality problem, which is known to be undecidable [1, 15].

We propose as open problems to determine the complexity of the following topological properties: Is $S_{\mathscr{C}}$ is a topological manifold? Is $S_{\mathscr{C}}$ contractible?

Acknowledgements

We thank the anonymous referees for clarifying an issue regarding the subtle difference between $\mathsf{P}^{\mathsf{NP}[\log]}$ and the corresponding function class, as well pointing out to us some references to natural problems for $\mathsf{P}^{\mathsf{NP}[\log]}$.

References

1. S.I. Adian. Unsolvability of certain algorithmic problems in the theory of groups (in Russian). *Trudy Moskov. Math. Obshch.*, 6:231–298, 1957.
2. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer-Verlag, 1998.
3. P. Bürgisser and F. Cucker. Counting complexity classes for numeric computations I: Semilinear sets. *SIAM J. Comp.*, 33:227–260, 2004.
4. A. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM J. Comp.*, 13:423–439, 1984.
5. F. Cucker and P. Koiran. Computing over the reals with addition and order: Higher complexity classes. *Journal of Complexity*, 11:358–376, 1995.
6. H. Fournier and P. Koiran. Are lower bounds easier over the reals?, In *Proc. 30th ACM STOC*, pages 507–513, 1998.
7. E. Hemaspaandra and L.A. Hemaspaandra and J. Rothe. Exact Analysis of Dodgson Elections: Lewis Carroll's 1876 Voting System is Complete for Parallel Access to NP. *Journal of the ACM*, pages 806–825, 1997.
8. L.G. Khachijan. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244:1093–1096, 1979. (In Russian, English translation in *Soviet Math. Dokl.*, 20:191–194, 1979.)
9. P. Koiran. Computing over the reals with addition and order. *Theoretical Computer Science*, 133:35–47, 1994.
10. M. W. Krentel. The complexity of optimization problems. In *Proc. 18th ACM Symp. on the Theory of Computing*, pages 79–86, 1986.
11. F. Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *J. ACM*, 31:668–676, 1984.
12. C.H. Papadimitriou. On the complexity of unique solutions. *J. ACM*, 31:392–400, 1984.
13. C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
14. C.H. Papadimitriou and S. Zachos. Two remarks on the power of counting. *Proc. 6th GI conference in Theoretical Computer Science*, Lecture Notes in Computer Science 145, Springer Verlag, pages 269–276, 1983.
15. M. Rabin. Recursive unsolvability of group theoretic problems. *Ann. of Math.*, 67(2):172–194, 1958.
16. I.R. Shafarevich. *Basic Algebraic Geometry.* Springer Verlag, 1974.
17. S. Smale. Mathematical problems for the next century. *Mathematical Intelligencer*, 20:7–15, 1998.
18. E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34:250–256, 1986.