THE COMPLEXITY OF SEMILINEAR PROBLEMS IN SUCCINCT REPRESENTATION

Peter Bürgisser, Felipe Cucker,
and Paulin Jacobé de Naurois

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 $\mathsf{P}^{\mathsf{NP}[\mathsf{log}]}$-completeness of deciding Zariski irreducibility, exhibits for the first time a problem with a geometric nature complete in this class.

Keywords. BSS additive model, semilinear sets, complete problems.
Subject classification. 68Q15.

1. Introduction

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 (see e.g., Dantzig & Eaves 1973; Ferrante & Rackoff 1979; Schechter 1998). 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 (1979) states that if the coefficients of these equalities and inequalities are integers then this problem can be solved in polynomial time in the Turing machine model. In other words, 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 (cf. Borgwardt 1982; Cheung *et al.* 2003; Smale 1983), 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 (1998) as one of the mathematical problems for the 21st century.

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 (1986), together with a suitable variant of Gaussian elimination (Bareiss 1968), 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 3): 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 $\mathrm{CSAT}_{\mathrm{add}}$ 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 (Blum *et al.* 1998). This is in contrast with the result by Tardos (1986) 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 $\mathrm{CSAT}_{\mathrm{add}}$ is not an isolated fact. It was recently shown in Bürgisser & Cucker (2003) that several other problems for semilinear sets in succinct representation are complete in some complexity class. Notably,

[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.

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

A final remark is relevant. Constant-free additive circuits can be described over $\{0,1\}$, i.e., as binary strings. Therefore, they can be given as input to Turing machines. In this way, all problems considered in this paper have a constant-free version fitting the classical complexity setting. We will also show that our completeness results in the additive model smoothly translate into completeness results in the usual Turing model when constant-free circuits are considered.

2. Main Results

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

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

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

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

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

ISOLATED$_{\mathrm{add}}$ (*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 $S_{\mathscr{C}}$.

$\text{EXISTISO}_{\text{add}}$ (*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 $S_{\mathscr{C}}$.

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

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

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

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

$\#\text{DISC}_{\text{add}}$ (*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).

$\text{REACH}_{\text{add}}$ (*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 $\mathcal{S}_{\mathscr{C}}$.

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

$\text{TORSION}_{\text{add}}$ (*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 (Shafarevich 1974). 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 & Zachos (1983) and consists of the decision problems that can be solved in polynomial time by $\mathcal{O}(\log n)$ queries to some NP language. It is known (Buss & Hay 1991; Hemachandra 1989) that equivalently,

$\mathsf{P}^{\mathsf{NP}[\log]}$ can also be characterized as the set of languages in P^{NP} whose queries are non adaptive. Several natural complete problems for $\mathsf{P}^{\mathsf{NP}[\log]}$ are known, see for instance Krentel (1986) and Hemaspaandra *et al.* (1997).

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 Bürgisser & Cucker (2003, §7). Also, the PAR_{add}-completeness of $\text{CONNECTED}_{\text{add}}$ closes a question left open therein.

3. Preliminaries

In this section we briefly 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 Bürgisser & Cucker (2003).

(1) The Euclidean norm in $\mathbb{R}^n$ induces a topology, called *Euclidean*, in $\mathbb{R}^n$. The same topology is induced by the maximum norm defined by $\|x\|_\infty := \max_i |x_i|$ for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. We will denote the closure of a subset $S \subseteq \mathbb{R}^n$ with respect to the Euclidean topology by $\overline{S}$.

We already defined a subset $S \subseteq \mathbb{R}^n$ to be *semilinear* if it is a Boolean combination of closed half-spaces. Following Shafarevich (1974), we define another, coarser, topology in $\mathbb{R}^n$, hereby restricting us to semilinear sets.

DEFINITION 3.1. *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.*

We remark that 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 $\mathsf{sg} : \mathbb{R} \to \{-1, 0, 1\}$ and $\mathsf{pos} : \mathbb{R} \to \{0, 1\}$ defined by

$$\mathsf{sg}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0 \end{cases} \qquad \mathsf{pos}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$

We extend these functions to $\mathbb{R}^n$ componentwise. By a *quadrant* of $\mathbb{R}^n$ we understand 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 fix some conventions concerning additive circuits. Such circuits are defined in many places, cf. Blum *et al.* (1998); Bürgisser & Cucker (2003); Koiran (1994). In order to simplify our proofs we will use a slightly different definition. An *additive circuit* is a directed acyclic graph whose nodes are of one of the following types: input, output, constant, addition, subtraction, and selection. While the first five types are as in the references above, we will consider the selection nodes of the circuits to have indegree 4 and compute, with input (v, a, b, c),

$$\text{if } v < 0 \text{ then } a, \text{ elsif } v = 0 \text{ then } b, \text{ else } c.$$

This is without loss of generality since one can pass from this form of circuit to the usual one (in which selection nodes have indegree 3 and compute "if $v < 0$ then a else b") in polynomial time and vice versa.

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 connected to the constants $a, b, c \in \{0, 1\}$. Such a circuit computes a function $F_{\mathscr{C}} : \mathbb{R}^n \to \{0, 1\}$ and decides the semilinear set $S_{\mathscr{C}} := \{x \in \mathbb{R}^n \mid F_{\mathscr{C}}(x_1, \ldots, x_n) = 1\}$. A semilinear set $S_{\mathscr{C}}$ represented this way will be said to be given in *succinct representation*.

Definition 3.2. *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. (That is, $\gamma_j = -1$ 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 will use Turing machines and the complexity theory built upon them, cf. Papadimitriou (1994). In particular, we will deal with the complexity classes of decision problems P, NP, PSPACE, and EXP as well as with the class $\#\mathsf{P}$ of counting problems or the class of functions FP computable in polynomial time.

We will also use additive machines (i.e., BSS machines over $\mathbb{R}$ which do not multiply or divide) as described in Blum *et al.* (1998, Chapter 18) or in Koiran (1994). For these machines, versions of the complexity classes mentioned above are also defined yielding the classes $\mathrm{P}_{\mathrm{add}}$, $\mathrm{NP}_{\mathrm{add}}$, $\mathrm{PAR}_{\mathrm{add}}$, $\mathrm{EXP}_{\mathrm{add}}$, $\#\mathrm{P}_{\mathrm{add}}$ 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 Blum *et al.* (1998, Chapter 18) and Bürgisser & Cucker (2003).

We already defined the problem $\mathrm{CS}\textsc{at}_{\mathrm{add}}$ and observed that it is $\mathrm{NP}_{\mathrm{add}}$-complete. Consider now the problems:

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

$\textsc{Dim}_{\mathrm{add}}$ (*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.

The problem $\mathrm{CBS}_{\mathrm{add}}$ is $\mathrm{NP}_{\mathrm{add}}$-complete (Cucker & Matamala 1996; Koiran 1994). The same is true for $\textsc{Dim}_{\mathrm{add}}$ (Bürgisser & Cucker 2003, Theorem 5.1, there k is assumed to be fixed, but the proof carries over easily).

We will use the completeness of both $\mathrm{CBS}_{\mathrm{add}}$ and $\textsc{Dim}_{\mathrm{add}}$ in our development. 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 $\{0,1\}^n$. This digital form of nondeterminism extends to the levels of the additive polynomial hierarchy and we will also use natural extensions of $\mathrm{CS}\textsc{at}_{\mathrm{add}}$ and $\mathrm{CBS}_{\mathrm{add}}$, which are complete in the lower levels of this hierarchy (again, see Blum *et al.* (1998, Ch. 18), Bürgisser & Cucker (2003) or Cucker & Koiran (1995)).

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. A series of results starting in Fournier & Koiran (1998, 2000), continued in Bürgisser & Cucker (2003), and relying on Meyer auf der Heide (1984), allow us to use standard discrete problems as basis for reductions yielding Turing-hardness results in the additive setting. More specifically, these results show the following (cf. Bürgisser & Cucker 2003, Theorem 4.1)

$$\Sigma^k_{\mathrm{add}} \subseteq \mathrm{P}^{\mathsf{\Sigma}^k}_{\mathrm{add}} \qquad \text{and} \qquad \mathrm{PAR}_{\mathrm{add}} = \mathrm{P}^{\mathsf{PSPACE}}_{\mathrm{add}}. \tag{3.3}$$

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

Lemma 3.4. *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* $\overline{D_\gamma}^Z \subseteq \overline{D_{\gamma'}}^Z$.

Proof. (i) This part is just a rephrasing of a well-known and important result due to Tardos (1986), based on polynomial time algorithms for linear programming over $\mathbb{Q}$, together with a polynomial time variant of Gaussian elimination over $\mathbb{Q}$ (Bareiss 1968).

(ii) Let $\mathscr{C}$ be an additive circuit with r selection gates and γ be a path of $\mathscr{C}$. Let $F_i(x) = \sum_{k=1}^n a_{ik}x_k + b_i$ denote the affine polynomial computed by $\mathscr{C}$ at the ith selection node. The coefficients a_{ik} are integers of bit-size polynomially bounded in the size of $\mathscr{C}$ and b_i is a real number (which is an integer linear combination of the machine constants), cf. Blum *et al.* (1998, Chapter 18).

We put $I_\gamma^+ = \{1 \leq i \leq r \mid \gamma_i = 1\}$, $I_\gamma^- = \{1 \leq i \leq r \mid \gamma_i = -1\}$, $I_\gamma = I_\gamma^+ \cup I_\gamma^-$, and $E_\gamma = \{1 \leq j \leq r \mid \gamma_j = 0\}$. Then the leaf set D_γ is the following convex set

$$\begin{aligned} D_\gamma = \{x \in \mathbb{R}^n \mid F_j(x) = 0 \text{ for all } j \in E_\gamma,\ F_i(x) > 0 \text{ for all } i \in I_\gamma^+, \\ \text{and } F_i(x) < 0 \text{ for all } i \in I_\gamma^-\}. \end{aligned}$$

If D_γ is not empty, then its Euclidean closure is given by

$$\begin{aligned} \overline{D_\gamma} = \{x \in \mathbb{R}^n \mid F_j(x) = 0 \text{ for all } j \in E_\gamma,\ F_i(x) \geq 0 \text{ for all } i \in I_\gamma^+, \\ \text{and } F_i(x) \leq 0 \text{ for all } i \in I_\gamma^-\}. \end{aligned}$$

Moreover, the Zariski closure $\overline{D_\gamma}^Z$ of D_γ is the affine hull of D_γ. Therefore

$$\overline{D_\gamma}^Z = \{x \in \mathbb{R}^n \mid F_j(x) = 0 \text{ for all } j \in E_\gamma\},$$

provided D_γ is not empty. Part (ii) follows now immediately.

(iii) It follows from the above that $\dim D_\gamma = \dim \overline{D_\gamma}^Z = n - \operatorname{rank} A$, where A denotes the integer matrix $(a_{jk})_{j\in E_\gamma, 1\leq k\leq n}$. It is known that the rank of an integer matrix can be computed in polynomial time by a Turing machine by a suitable variant of Gaussian elimination, cf. Bareiss (1968). In particular, such a computation can be performed by an additive machine in polynomial time. This shows Part (iii).

(iv) Note that $\overline{D_\gamma}^Z \subseteq \overline{D_{\gamma'}}^Z$ is equivalent to $\dim(\overline{D_\gamma}^Z \cap \overline{D_{\gamma'}}^Z) = \dim \overline{D_\gamma}^Z$. Moreover, by the proof of Part (ii), we can compute equations for $\overline{D}_\gamma^Z \cap \overline{D_{\gamma'}}^Z$ in time polynomial in the size of $\mathscr{C}$. The claim follows now from Part (iii). $\square$

Remark 3.5. *If we have the a priori information that D_γ is nonempty, then the polynomial time algorithms to solve the above tasks (ii)–(iv) do not rely on Tardos' algorithm. In the proofs of all our results that follow, it is enough to use the fact that $D_\gamma \neq \emptyset$ can be (trivially) certified in* $\mathrm{NP}_{\mathrm{add}}$. *So we do not rely on Tardos' algorithm.*

4. Properties for the Euclidean Topology

In this section we study the complexity of several topological properties of semilinear sets, where the considered topology is the Euclidean one.

4.1. Euclidean Adherence, Closedness, and Denseness.

Proposition 4.1. *The problem* $\mathrm{EA{\small DH}}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.*

Proof. We first exhibit an $\mathrm{NP}_{\mathrm{add}}$ algorithm solving $\mathrm{EA{\small DH}}_{\mathrm{add}}$. Let $\mathscr{C}$ be a decision circuit with n input gates and r selection gates. We have

$$\overline{S_{\mathscr{C}}} = \overline{\bigcup_{\gamma\in\mathcal{A}_{\mathscr{C}}} D_\gamma} = \bigcup_{\gamma\in\mathcal{A}_{\mathscr{C}}} \overline{D_\gamma}.$$

Hence $x \in \overline{S_{\mathscr{C}}}$ iff $\exists \gamma \in \mathcal{A}_{\mathscr{C}}\ x \in \overline{D_\gamma}$. By Lemma 3.4, the property $x \in \overline{D_\gamma}$ can be tested in $\mathrm{P}_{\mathrm{add}}$. This proves the membership of $\mathrm{EA{\small DH}}_{\mathrm{add}}$ to $\mathrm{NP}_{\mathrm{add}}$.

For proving the hardness, we reduce $\mathrm{CBS}_{\mathrm{add}}$ to $\mathrm{EA{\small DH}}_{\mathrm{add}}$. Assume $\mathscr{C}$ is a decision circuit with n input gates. Consider a circuit $\mathscr{C}'$ computing the function

$$G_{\mathscr{C}}\colon \mathbb{R}^n \to \{0,1\},\ x \mapsto F_{\mathscr{C}}(\mathsf{pos}(x)). \tag{4.2}$$

The mapping $\mathscr{C} \mapsto (\mathscr{C}', 0)$ reduces $\mathrm{CBS}_{\mathrm{add}}$ to $\mathrm{EA{\small DH}}_{\mathrm{add}}$. 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 $0 \in \overline{S_{\mathscr{C}'}}$. □

Proposition 4.3. *The problem* $\mathrm{EC{\small LOSED}}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

Proof. We first prove that $\mathrm{EC{\small LOSED}}_{\mathrm{add}}$ belongs to $\mathrm{coNP}_{\mathrm{add}}$. Let a decision circuit $\mathscr{C}$ with n input gates be given. Then, $S_{\mathscr{C}}$ is closed if and only if

$$\begin{aligned}\forall x \in \mathbb{R}^n\ \big(x \in \overline{S_{\mathscr{C}}} \Rightarrow x \in S_{\mathscr{C}}\big) &\iff \forall x \in \mathbb{R}^n\ \big((\exists \gamma \in \mathcal{A}_{\mathscr{C}}\ x \in \overline{D_\gamma}) \Rightarrow x \in S_{\mathscr{C}}\big)\\ &\iff \forall x \in \mathbb{R}^n\ \big(\forall \gamma \in \mathcal{A}_{\mathscr{C}}\ x \notin \overline{D_\gamma}\big) \vee x \in S_{\mathscr{C}}.\end{aligned}$$

Since the predicate $x \notin \overline{D_\gamma}$ can be checked in $\mathrm{P}_{\mathrm{add}}$ by Lemma 3.4 we are done.

For proving hardness, we reduce $\mathrm{CBS}_{\mathrm{add}}$ to the complement of $\mathrm{E}\textsc{Closed}_{\mathrm{add}}$ with a reduction similar to the one of Proposition 4.1. Assume $\mathscr{C}$ is a decision circuit with n input gates. Consider a circuit $\mathscr{C}'$ computing the following

input $x \in \mathbb{R}^n$
if $x = 0$ REJECT else if $\mathsf{pos}(x) \in S_{\mathscr{C}}$ ACCEPT else REJECT

Clearly, $S_{\mathscr{C}} \cap \{0,1\}^n = \emptyset$ implies $S_{\mathscr{C}'} = \emptyset$, which is closed. Conversely, if $S_{\mathscr{C}} \cap \{0,1\}^n \neq \emptyset$, then $S_{\mathscr{C}'}$ is not closed since $0 \in \overline{S_{\mathscr{C}'}}$ and $0 \notin S_{\mathscr{C}'}$. □

Proposition 4.4. *The problem* $\mathrm{E}\textsc{Dense}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

Proof. Given a decision circuit $\mathscr{C}$ with n input gates, $S_{\mathscr{C}}$ is dense in $\mathbb{R}^n$ if and only if its complement has dimension strictly less than n. This is in $\mathrm{coNP}_{\mathrm{add}}$ since $\textsc{Dim}_{\mathrm{add}}$ is in $\mathrm{NP}_{\mathrm{add}}$ by Bürgisser & Cucker (2003, Theorem 5.1).

To show the hardness we reduce $\mathrm{CBS}_{\mathrm{add}}$ to the complement of $\mathrm{E}\textsc{Dense}_{\mathrm{add}}$ by assigning to a decision circuit $\mathscr{C}$ a circuit $\mathscr{C}''$ computing the function $x \mapsto 1 - G_{\mathscr{C}}(x)$, where $G_{\mathscr{C}}$ is the function introduced in (4.2). □

4.2. Unboundedness and Compactness.

Proposition 4.5. *The problem* $\textsc{Unbounded}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.*

Proof. For proving the hardness we reduce $\mathrm{C}\textsc{Sat}_{\mathrm{add}}$ to $\textsc{Unbounded}_{\mathrm{add}}$. Let $\mathscr{C}$ be a decision circuit. Define $\mathscr{C}'$ by adding a dummy variable to $\mathscr{C}$. Then $S_{\mathscr{C}'}$ is a cylinder of base $S_{\mathscr{C}}$ satisfying that $S_{\mathscr{C}}$ is non-empty if and only if $S_{\mathscr{C}'}$ is so, and in this case the latter is unbounded.

For the membership, assume that $S_{\mathscr{C}} \subseteq \mathbb{R}^n$ is bounded. Then, every accepting path γ defines a bounded leaf set D_γ and $\overline{S_{\mathscr{C}}} = \bigcup_{\gamma \in \mathcal{A}_{\mathscr{C}}} \overline{D_\gamma}$. Denote by y a point of $\overline{S_{\mathscr{C}}}$ which is at the greatest distance from the origin. Then y is a vertex of one of the polyhedra $\overline{D_\gamma}$. This vertex is therefore defined by n equalities: y is the unique solution of a system $Ay = b$, where $A = (a_{ij})$ is a $n \times n$ integer matrix and $|a_{ij}| \leq 2^{p(n)}$ for some polynomial p. The b_i are integer linear combinations of the constants $c_1, \ldots, c_k \in \mathbb{R}$ of the circuit $\mathscr{C}$ defining S, hence $b_i = \sum_{j=1}^k \beta_{ij} c_j$ with $\beta_{ij} \in \mathbb{Z}$ and $|\beta_{ij}| \leq 2^{p(n)}$. Therefore, $|b_i| \leq kc2^{p(n)}$, where $c = \max |c_j|$.

By Cramer's rule we have $y_i = \det A_i^b / \det A$, where A_i^b denotes the matrix obtained by replacing in A the ith column by b. Since A is invertible and $a_{ij} \in \mathbb{Z}$, we have $|\det A| \geq 1$. Denote by $(A)_{ij}$ the submatrix of A obtained by removing the ith column and the jth row. Developing $\det A_i^b$ along the ith column, we get

$$\det A_i^b = \sum_{j=1}^n (-1)^{i+j} b_j \det A_{ij}.$$

Taking into account that $|\det A_{ij}| \le n! 2^{np(n)}$ we obtain

$$|y_i| \le |\det A_i^b| \le kcn2^{p(n)} n! 2^{np(n)} =: B(n).$$

This bound is clearly computable in $\mathrm{FP}_{\mathrm{add}}$.

An $\mathrm{NP}_{\mathrm{add}}$ algorithm for $\textsc{Unbounded}_{\mathrm{add}}$ now easily follows. Given a circuit $\mathscr{C}$, guess a point $y \in \mathbb{R}^n$ with $\|y\|_\infty > B(n)$. Then accept if and only if $\mathscr{C}(y) = 1$. □

Proposition 4.6. *The problem* $\textsc{Compact}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

Proof. Membership follows from the membership of $\textsc{Unbounded}_{\mathrm{add}}$ to $\mathrm{NP}_{\mathrm{add}}$ and of $\textsc{EClosed}_{\mathrm{add}}$ to $\mathrm{coNP}_{\mathrm{add}}$. Hardness follows from the reduction of Proposition 4.3. □

4.3. Isolated points and Local Dimension.

Proposition 4.7. *The problem* $\textsc{Isolated}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

Proof. Membership easily follows from the equivalence

$$x \text{ not isolated in } S \Longleftrightarrow x \notin S \ \vee\ x \in \overline{S \setminus \{x\}}$$

and the membership of $\textsc{EAdh}_{\mathrm{add}}$ to $\mathrm{NP}_{\mathrm{add}}$. Hardness follows from Proposition 4.1 and the equivalence

$$x \in \overline{S} \Longleftrightarrow x \in S \ \vee\ x \text{ not isolated in } S \cup \{x\},$$

which reduces from $\textsc{EAdh}_{\mathrm{add}}$ to the complement of $\textsc{Isolated}_{\mathrm{add}}$. □

Proposition 4.8. *The problem* $\textsc{ExistIso}_{\mathrm{add}}$ *is* Σ^2_{add}*-complete.*

Proof. The membership to Σ^2_{add} trivially follows from Proposition 4.7. For the hardness, we will use the problem $\Sigma^2\mathrm{CBS}_{\mathrm{add}}$ consisting of deciding, given a decision circuit $\mathscr{C}$ with $n+m$ input gates, whether

$$\exists x \in \{-1,1\}^n \ \forall y \in \{-1,1\}^m \ (x,y) \in S_{\mathscr{C}}.$$

This is a Σ^2_{add}-complete problem and we will reduce it to $\textsc{ExistIso}_{\mathrm{add}}$. To do so, let $\mathscr{C}'$ be the decision circuit computing the following

input $(x,y) \in \mathbb{R}^{n+m}$
if $x \notin \{-1,1\}^n$ REJECT else if $y = 0$ ACCEPT
else if $\exists j \le m$ such that $y_j = 0$ REJECT
else if $(x, \mathsf{sg}(y)) \notin S_{\mathscr{C}}$ ACCEPT else REJECT

By definition, all points in $S_{\mathscr{C}'}$ have as x-component a vertex of an n-dimensional hypercube. Over each one of these points, say x_0, lies a space $\mathbb{R}^m$ corresponding to the y coordinates. The origin of this space is in $S_{\mathscr{C}'}$. The possible other points of $S_{\mathscr{C}'}$ in this space must lie in quadrants. Moreover, a point in this space lies in a quadrant if and only if $\exists y \in \{-1,1\}^m\ (x_0,y) \notin S_{\mathscr{C}}$. That is, $(x_0,0)$ is isolated in $S_{\mathscr{C}'}$ if and only if $\forall y \in \{-1,1\}^m\ (x_0,y) \in S_{\mathscr{C}}$. Since the only possible isolated points in $S_{\mathscr{C}'}$ are those with the form $(x,0)$ with $x \in \{-1,1\}^n$ it follows that $S_{\mathscr{C}'}$ has isolated points if and only if $\mathscr{C} \in \Sigma^2\mathrm{CBS}_{\mathrm{add}}$. □

Our next completeness result, Corollary 4.11 below, is about counting problems and complexity classes. We briefly remind the reader of the main notions involved.

Definition 4.9. *Given a set $A \subseteq \mathbb{R}^\infty$ and a polynomial p, we define the functions $\#^p_A, \mathrm{D}\#^p_A : \mathbb{R}^\infty \to \mathbb{N} \cup \{\infty\}$ which associate to $x \in \mathbb{R}^n$ the cardinalities*

$$\begin{aligned} \#^p_A(x) &= |\{y \in \mathbb{R}^{p(n)} \mid (x,y) \in A\}|, \\ \mathrm{D}\#^p_A(x) &= |\{y \in \{0,1\}^{p(n)} \mid (x,y) \in A\}|, \end{aligned}$$

respectively. If $\mathcal{C} \subseteq 2^{\mathbb{R}^\infty}$ is a complexity class of decision problems, we define

$$\begin{aligned} \# \cdot \mathcal{C} &= \{\#^p_A \mid A \in \mathcal{C} \text{ and } p \text{ a polynomial}\}, \\ \mathrm{D}\# \cdot \mathcal{C} &= \{\mathrm{D}\#^p_A \mid A \in \mathcal{C} \text{ and } p \text{ a polynomial}\}. \end{aligned}$$

When $\mathcal{C} = \mathrm{P}_{\mathrm{add}}$ we write $\#\mathrm{P}_{\mathrm{add}}$ instead of $\# \cdot \mathrm{P}_{\mathrm{add}}$. These definitions mimic similar definitions in the discrete setting. The class $\#\mathsf{P}$ was introduced by Valiant (1979a,b) in seminal papers. 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 (1991) proving that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$, i.e., that $\#\mathsf{P}$ has at least the power of the polynomial hierarchy. This result was extended by Toda & Watanabe (1992) who showed that $\# \cdot \mathsf{PH} \subseteq \mathsf{FP}^{\#\mathsf{P}}$.

In Bürgisser & Cucker (2003, Corollary 4.6) the result of Toda & Watanabe (1992) was extended to show that $\mathrm{D}\# \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$. Using the same ideas, it is not hard to further extend this as follows.

THEOREM 4.10. *We have* $\# \cdot \mathrm{PH}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$.

PROOF. The proof of the inclusion is analogous to the one of Bürgisser & Cucker (2003, Theorem 4.7) and we therefore only sketch it. There, it was shown that if φ is a function in $\#\mathrm{P}_{\mathrm{add}}$ that takes only finite values, then $\varphi \in \mathrm{D}\# \cdot \mathrm{NP}_{\mathrm{add}}$. Moreover, for given $x \in \mathbb{R}^\infty$, one can test in $\mathrm{NP}_{\mathrm{add}}$ whether $\varphi(x)$ is infinite, cf. Bürgisser & Cucker (2003, Lemma 3.4).

The same arguments show that if $\varphi \in \Sigma_{\mathrm{add}}^k$, $k > 0$, then we can test in Σ_{add}^k whether $\varphi(x)$ is infinite. Moreover, if φ takes only finite values, then $\varphi \in \mathrm{D}\# \cdot \Sigma_{\mathrm{add}}^k$. Combining these two arguments proves the inclusion.

The equality $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$ is from Bürgisser & Cucker (2003, Theorem 4.7). $\square$

COROLLARY 4.11. *The problem* $\#\textsc{Iso}_{\mathrm{add}}$ *is Turing-complete in* $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$.

PROOF. The membership follows from Proposition 4.7 and Theorem 4.10.

For proving the hardness, since $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}} = \mathrm{FP}_{\mathrm{add}}^{\#\mathsf{P}}$, it is enough to reduce $\#\textsc{Sat}$ to $\#\textsc{Iso}_{\mathrm{add}}$. To do so, assume ϕ is an input for $\#\textsc{Sat}$, i.e., a Boolean formula in conjunctive normal form. It is easy to compute from ϕ a decision circuit $\mathscr{C}$ which accepts only the subset of $\{0,1\}^n$ consisting of the satisfying assignments of ϕ. Then, by definition, every point in $S_{\mathscr{C}}$ is isolated, and the number of points in $S_{\mathscr{C}}$ is the number of satisfying assignments for ϕ. $\square$

The *local dimension* of a semilinear set $S \subseteq \mathbb{R}^n$ at $x \in S$ is defined as $\dim_x S := \min_{r>0} \dim(S \cap \mathcal{B}_r(x))$, where $\mathcal{B}_r(x)$ denotes the open ball of radius r centered at x.

PROPOSITION 4.12. *The problem* $\textsc{LocDim}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.*

PROOF. The membership follows from the observation that, for $x \in S$,

$$\dim_x S \geq d \iff \exists \gamma \in \mathcal{A}_{\mathscr{C}}\ x \in \overline{D_\gamma} \wedge \dim D_\gamma \geq d.$$

The predicates $x \in \overline{D_\gamma}$ and $\dim D_\gamma \geq d$ can be checked in $\mathrm{P}_{\mathrm{add}}$ by Lemma 3.4.

The hardness follows from the following equivalence

$$x \text{ isolated in } S \iff x \in S \wedge \dim_x S < 1$$

and from Proposition 4.7. $\square$

4.4. Continuity and Counting Points of Discontinuity.

PROPOSITION 4.13. *The problem* $\mathrm{LocCont}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

PROOF. Let us first focus on the membership. Given a decision circuit $\mathscr{C}$, the local continuity of $F_{\mathscr{C}}$ in a point $x \in \mathbb{R}^n$ can be expressed in the following way:

$$\forall \epsilon > 0 \; \exists \eta > 0 \; \forall y \in \mathbb{R}^n \; \big(\|x - y\|_\infty < \eta \Rightarrow \|F_{\mathscr{C}}(x) - F_{\mathscr{C}}(y)\|_\infty < \epsilon\big). \tag{4.14}$$

For $\epsilon > 0$ consider the following semilinear set

$$S_x^\epsilon := \{y \in \mathbb{R}^n \mid \|F_{\mathscr{C}}(x) - F_{\mathscr{C}}(y)\|_\infty < \epsilon\}.$$

The local continuity of $F_{\mathscr{C}}$ in x can then be expressed as follows:

$$\forall \epsilon > 0 \quad x \notin \overline{(\mathbb{R}^n \setminus S_x^\epsilon)}. \tag{4.15}$$

Note that, given ϵ and x, a circuit for $\mathbb{R}^n \setminus S_x^\epsilon$ can be computed in polynomial time. In addition, $\mathrm{EAdh}_{\mathrm{add}}$ is in $\mathrm{NP}_{\mathrm{add}}$ by Proposition 4.1. Therefore, (4.15) can be decided in $\mathrm{coNP}_{\mathrm{add}}$ and with it, the continuity of $F_{\mathscr{C}}$ at $x \in \mathbb{R}^n$.

For proving the hardness, consider the reduction of Proposition 4.3. The function $F_{\mathscr{C}'}$ is continuous at the origin if and only if $S_{\mathscr{C}} = \emptyset$. Therefore it reduces $\mathrm{CBS}_{\mathrm{add}}$ to the complement of $\mathrm{LocCont}_{\mathrm{add}}$. □

PROPOSITION 4.16. *The problem* $\mathrm{Cont}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete.*

PROOF. The membership is a consequence of Proposition 4.13: it suffices to check the local continuity at all points. For the hardness, we reduce $\mathrm{CBS}_{\mathrm{add}}$ to the complement of $\mathrm{Cont}_{\mathrm{add}}$. To a decision circuit $\mathscr{C}$ with n input gates we assign a circuit $\mathscr{C}'$ computing the indicator function f of $S_{\mathscr{C}} \cap \{0,1\}^n$ in $\mathbb{R}^n$. Then $S_{\mathscr{C}} \cap \{0,1\}^n$ is empty iff f is continuous. □

PROPOSITION 4.17. *The problem* $\#\mathrm{Disc}_{\mathrm{add}}$ *is Turing-complete in* $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$.

PROOF. The upper bound follows from Proposition 4.13 and Theorem 4.10. For the lower bound note that the reduction used in the proof of Proposition 4.16 is parsimonious. □

4.5. Surjectivity.

PROPOSITION 4.18. *The problem* SURJ_{add} *is* Π^2_{add}*-complete.*

PROOF. The membership follows from the definition of surjectivity. Given an additive circuit $\mathscr{C}$, $F_{\mathscr{C}}$ is surjective if and only if

$$\forall y \in \mathbb{R}^m \; \exists x \in \mathbb{R}^n \; F_{\mathscr{C}}(x) = y.$$

For the hardness, we reduce the Π^2_{add}-complete problem $\Pi^2\text{CSAT}_{\text{add}}$ to SURJ_{add}. We recall, the former is the problem of deciding, given a decision circuit $\mathscr{C}$ with $n+m$ input gates, whether $\forall y \in \mathbb{R}^m \; \exists x \in \mathbb{R}^n \; F_{\mathscr{C}}(x,y) = 1$.

For the reduction, we associate to $\mathscr{C}$ another circuit $\mathscr{C}'$ computing the following

input $(a, x_1, x_2, (y_1, \ldots, y_m)) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$
define $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$ and denote $\underline{0} := (0 \ldots, 0) \in \mathbb{R}^m$
if $y = \underline{0}$ then
  if $F_{\mathscr{C}}(x_1, \underline{0}) = 1$ return $\underline{0}$ else return $(-1, \ldots, -1) \in \mathbb{R}^m$
else if $F_{\mathscr{C}}(x_1, (-y_1, y_2, \ldots, y_m)) = 1$ and $F_{\mathscr{C}}(x_2, (y_1, y_2, \ldots, y_m)) = 1$ and $a > 0$
  then return $(|y_1|, y_2, \ldots, y_m)$ else return $(-|y_1|, y_2, \ldots, y_m)$

We now show that $F_{\mathscr{C}'} : \mathbb{R}^{1+2n+m} \to \mathbb{R}^m$ is surjective if and only if $\mathscr{C} \in \Pi^2\text{CSAT}_{\text{add}}$. Assume $F_{\mathscr{C}'}$ is surjective. Then:

1. There exists $(a, x_1, x_2, y) \in \mathbb{R}^{1+2n+m}$ such that $F_{\mathscr{C}'}(a, x_1, x_2, y) = \underline{0}$. This occurs only when $y = \underline{0}$ and $F_{\mathscr{C}}(x_1, \underline{0}) = 1$. Hence $\exists x \in \mathbb{R}^n F_{\mathscr{C}}(x, \underline{0}) = 1$.

2. For all $y \in \mathbb{R}^m \setminus \{\underline{0}\}$, there exists $(a, x_1, x_2, z) \in \mathbb{R}^{1+2n+m}$ such that $F_{\mathscr{C}'}(a, x_1, x_2, z) = (|y_1|, y_2, \ldots, y_m)$. This occurs only when $a > 0$ and $|z_1| = |y_1|$, $z_i = y_i$ for $2 \le i \le m$ and $F_{\mathscr{C}}(x_1, (-z_1, z_2, \ldots, z_m)) = 1$ and $F_{\mathscr{C}}(x_2, (z_1, z_2, \ldots, z_m)) = 1$. Then, either $F_{\mathscr{C}}(x_1, y) = 1$ or $F_{\mathscr{C}}(x_2, y) = 1$.

It follows that $\mathscr{C} \in \Pi^2\text{CSAT}_{\text{add}}$. Assume now that $\mathscr{C} \in \Pi^2\text{CSAT}_{\text{add}}$. Then

1. There exists $x \in \mathbb{R}^n$ such that $F_{\mathscr{C}}(x, \underline{0}) = 1$. It follows $F_{\mathscr{C}'}(0, x, 0, \underline{0}) = \underline{0}$.

2. For all $y = (y_1, \ldots, y_m) \in \mathbb{R}^m \setminus \{\underline{0}\}$ there exist points $x_1, x_2 \in \mathbb{R}^n$ such that $F_{\mathscr{C}}(x_1, (-y_1, y_2, \ldots, y_m)) = 1$ and $F_{\mathscr{C}}(x_2, (y_1, y_2, \ldots, y_m)) = 1$. In this case, we have $F_{\mathscr{C}'}(1, x_1, x_2, y) = (|y_1|, y_2, \ldots, y_m)$ and $F_{\mathscr{C}'}(-1, x_1, x_2, y) = (-|y_1|, y_2, \ldots, y_m)$.

It follows that $F_{\mathscr{C}'}$ is surjective. □

5. Problems of Connectivity

5.1. Reachability and Connectedness. While the following result is already proven for semilinear sets of arbitrary dimension in Bürgisser & Cucker (2003), we need an alternative and simpler argument for proving Theorem 5.2 later on. Our graph-theoretic arguments are largely inspired by a similar result stated for graphs in Chandra *et al.* (1984).

LEMMA 5.1. *The problem* $\mathrm{REACH}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-complete under Turing reductions. The same holds when restricted to problems in* $\mathbb{R}^3$.

PROOF. For the membership, we refer the reader to Bürgisser & Cucker (2003) and therefore focus on the hardness. We remarked in Equation (3.3) that $\mathrm{PAR}_{\mathrm{add}} = \mathrm{P}_{\mathrm{add}}^{\mathsf{PSPACE}}$. To obtain Turing-hardness for $\mathrm{REACH}_{\mathrm{add}}$ it is therefore enough to prove that $\mathrm{REACH}_{\mathrm{add}}$ is PSPACE-hard.

Before going into the details, we note here that the general idea is to reduce the computation of a PSPACE Turing machine to an instance of $\mathrm{REACH}_{\mathrm{add}}$. We use three dimensions since we need one each for the configuration and the step counter and the third is necessary to create sufficient space for expressing connectedness of vertices by linear inequalities.

Let $L \subseteq \{0,1\}^*$ be any language in PSPACE. Then L is decided by a single tape deterministic Turing machine M with polynomial space bound function $p(n)$. Denote by Σ the alphabet of M, and by Q its set of states. For a fixed input length n, a *configuration* of M is an element c in the set $C_n = Q \times \{1,\ldots,p(n)\} \times \Sigma^{p(n)}$. We will identify C_n with the set $\{1,2,\ldots,T\}$ for a suitable T by interpreting a configuration c as a natural number written in base $|\Sigma|$. To an input $x \in \{0,1\}^n$, we also assign an initial configuration $i(x) \in C_n$. Without loss of generality, we may assume that there are unique accepting and rejecting configurations c_A and $c_R \in C_n$, respectively. Since M is deterministic, it accepts or rejects an input x in less than T computation steps. We may assume that after reaching c_A or c_R, the machine enters an infinite loop.

In the following we will use the notation $[T] := \{0,1,\ldots,T\}$. Consider the undirected graph $G_n = (V_n, E_n)$, where

$$V_n = (C_n \times [T]) \cup \{(c_A, T+1), (c_R, T+1)\}$$

and, for $t < T$,

$$\begin{aligned} \{(c,t),(c',t+1)\} \in E_n \quad &\text{iff} \quad c' \text{ is the next configuration of } M \text{ from } c \\ \{(c,T),(c_A,T+1)\} \in E_n \quad &\text{iff} \quad c = c_A \\ \{(c,T),(c_R,T+1)\} \in E_n \quad &\text{iff} \quad c \neq c_A. \end{aligned}$$

Clearly, $x \in L$ if and only if there exists a path from $(i(x), 0)$ to $(c_A, T+1)$ in G_n. We claim that the graph G_n satisfies the following properties:

(i) G_n can be succinctly described, i.e., there exists a Boolean circuit of size polynomial in n deciding whether two given vertices are linked by an edge in G_n.

(ii) G_n is a forest with two trees, which can be rooted at the vertices $(c_A, T+1)$ and $(c_R, T+1)$.

(iii) G_n can be embedded in $\mathbb{R}^3$ as a semilinear set representable by a decision circuit of size polynomial in n.

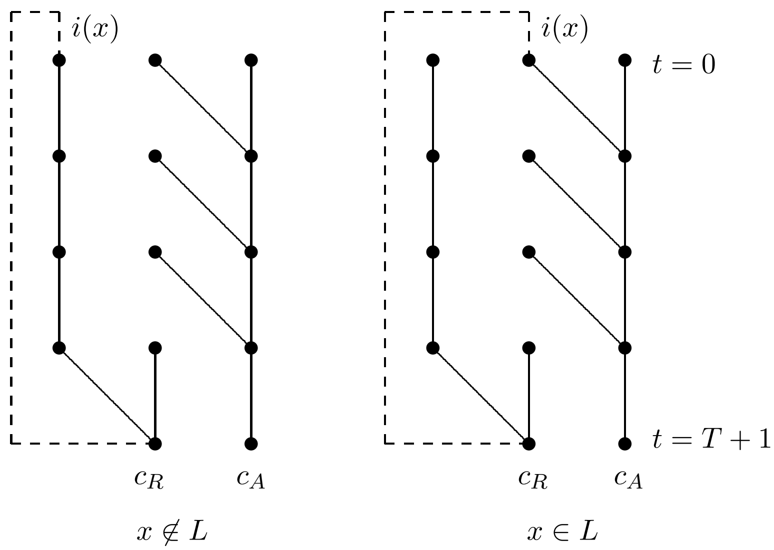

Figure 5.1: Two graphs G_n, with $T = 3$, for the cases $x \notin L$ and $x \in L$ respectively.

As the Claim (i) is obvious, we prove Claim (ii). Since M is deterministic, each configuration $c \in C_n$ has a unique next configuration $c' \in C_n$. Therefore, for $t < T$, each vertex (c, t) is connected to a unique vertex $(c', t+1)$. This implies that every connected component of G_n is a tree. Since any node (c, T) is connected to either $(c_A, T+1)$ or $(c_R, T+1)$ (but not to both), the graph G_n has exactly two connected components. See Figure 5.1 for an illustration (note, though, the dashed lines are not edges in G_n; they will be referred to in the proof of Theorem 5.2 below). For the rest of this proof, the two vertices

$(c_A, T+1)$ and $(c_R, T+1)$ will be called the roots of these two connected components.

We finally prove (iii). Define the layer t of G_n to be the set $\{(c,t) \mid c \in C_n\}$. It follows from (ii) that the edges of G_n link only vertices of two consecutive layers. Let us focus first on the geometrical representation in $\mathbb{R}^3$ of two consecutive layers, corresponding to vertices of the form (c,t) and $(c', t+1)$ for a given $t \leq T$.

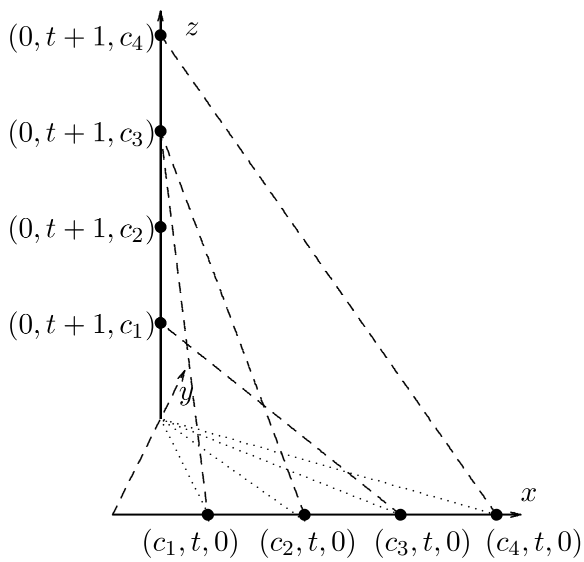

Figure 5.2: $\mathcal{S}_n(t)$ with $T = 4$ and t even.

Consider the set $\mathcal{S}_n(t) \subseteq \mathbb{R}^3$ of Figure 5.2 defined as follows. For t even, the vertices (c,t) are represented by the points $(c,t,0) \in \mathbb{R}^3$ and the vertices $(c', t+1)$ are represented by the points $(0, t+1, c') \in \mathbb{R}^3$. When (c,t) and $(c', t+1)$ are linked by an edge in G_n, their corresponding representations in $\mathcal{S}_n(t)$ are connected by a line segment. The same idea applies for t odd, with vertices (c,t) represented by the points $(0,t,c) \in \mathbb{R}^3$ and vertices $(c, t+1)$ represented by the points $(c, t+1, 0) \in \mathbb{R}^3$.

Note that when t is even, $\mathcal{S}_n(t)$ lies in the tetrahedron with vertices $(1,t,0)$, $(T,t,0)$, $(0,t+1,1)$, and $(0,t+1,T)$ (see Figure 5.2). This tetrahedron is defined by the inequalities $y-t-1+x \geq 0$, $y-t-z \leq 0$, $T(y-t-1)+x \geq 0$, and $T(y-t)-z \leq 0$.

It is easy to check that the tridimensional representations of two different edges in G_n intersect only at the end points. Indeed, let e_1 and e_2 be two such

```
input (x, y, z) ∈ [0, T + 1]^3
if y = T + 1
   if x ∈ {c_A, c_R} ∧ z = 0 then ACCEPT else REJECT
else
compute t = ⌊y⌋ # 0 ≤ t ≤ T, t ≤ y < t + 1
if t is even then
   if y = t ∧ x ∈ {0, 1, ..., T} ∧ z = 0 then ACCEPT
      # (x, y, z) represents a vertex of G_n
   else # t < y < t + 1
(i)    if y − t − 1 + x < 0 ∨ y − t − z > 0 then REJECT
       else
(ii)   compute c_t = max{c ∈ {1, ..., T} :   c(y − t − 1) + x ≥ 0}
       compute c_{t+1} = max{c ∈ {1, ..., T} :   c(y − t) − z ≤ 0}
(iii)  if c_t(y − t − 1) + x = 0 ∧ c_{t+1}(y − t) − z = 0 ∧ {(c_t, t), (c_{t+1}, t + 1)} ∈ E_n
       then ACCEPT else REJECT
else
# the case where t is odd is treated similarly
```

Figure 5.3: Algorithm for deciding S_n.

edges and s_1 and s_2 be their corresponding representations in $\mathbb{R}^3$. Without loss of generality we assume that the orthogonal projections of s_1, s_2 onto the plane $z = 0$ lie in the strip $\mathbb{R} \times [t, t+1]$ and that t is even. Then these orthogonal projections do not intersect in the open strip $\mathbb{R} \times (t, t+1)$ (since any configuration has a unique next configuration). It follows that s_1 and s_2 do not intersect above this open strip, i.e., they may only intersect at their end points.

Define now the set $\mathcal{S}_n \subseteq \mathbb{R}^3$ as the union of the $\mathcal{S}_n(t)$ for $t \in [T]$. This set is the desired tridimensional representation of G_n. Without loss of generality we will assume that T is odd.

We claim that the algorithm given in Figure 5.3 decides $\mathcal{S}_n$. The detailed proof of this claim is left to the reader; let us just make some comments. First note that Step (i) guarantees that the maxima defining c_t and c_{t+1} exist. The point (x, y, z) lies on a line segment connecting $(c, t, 0)$ with $(0, t+1, c')$ for $c, c' \in [T]$ iff $c = c_t$, $c' = c_{t+1}$, $c_t(y - t - 1) + x = 0$, and $c_{t+1}(y - t) - z = 0$. Moreover, this line segment represents an edge of G_n iff $\{(c_t, t), (c_{t+1}, t+1)\}$ is an edge of G_n.

In order to implement the algorithm as an additive circuit of size polynomial in n, we use the binary representation for the occurring natural numbers $c \in [T]$. Then it is possible to compute the product of c with a real number y using additions only (fast exponentiation). Note that the bit size of T is polynomially bounded in n.

Summarizing, the above algorithm can be implemented by an additive circuit $\mathscr{C}_n$ of size polynomial in n. Moreover, we have $x \in L$ iff the images of $(i(x), 0)$ and $(c_A, T+1)$ are connected in $\mathcal{S}_n$. Hence, the mapping $x \mapsto (\mathscr{C}_n, (i(x), 0), (c_A, T+1))$ reduces L to $\text{REACH}_{\text{add}}$. □

THEOREM 5.2. *The problem* $\text{CONNECTED}_{\text{add}}$ *is* PAR_{add}*-complete under Turing reductions. The same holds when restricted to problems in* $\mathbb{R}^3$.

PROOF. The membership follows from Bürgisser & Cucker (2003, Theorem 5.19) where it is proved that the computation of the 0th Betti number $b_0(S_{\mathscr{C}})$ is FPAR_{add}-complete. Note that $S_{\mathscr{C}}$ is connected if and only if $b_0(S_{\mathscr{C}}) = 1$.

The hardness proof relies heavily on the construction of the graph G_n in the proof of Lemma 5.1. Consider the graph G'_n derived from G_n by adding an edge between $(i(x), 0)$ and $(c_R, T+1)$ (the dashed edge in Figure 5.1). Then G'_n is connected if and only if $i(x)$ and c_A lie in the same connected component of G_n. This is the case iff $x \in L$.

Now, from the semilinear set $\mathcal{S}_n \subseteq \mathbb{R}^3$ of Lemma 5.1 representing G_n, it is easy to construct a semilinear set representing G'_n by realizing the additional edge by a chain of line segments. □

5.2. Torsion-free Homology. Algebraic topology studies topological spaces by assigning to them various algebraic objects in a functorial way. In particular, homeomorphic (or even homotopy equivalent) spaces lead to isomorphic algebraic objects. General references for algebraic topology are Hatcher (2002); Munkres (1984) and a survey on recent applications can be found in Dey *et al.* (1999). Typical examples of such algebraic objects assigned to a space X are the (singular) homology groups $H_k(X; \mathbb{Z})$. Those are abelian groups, which are finitely generated if X is a finite cell complex (e.g., a semilinear set). Let $T_k(X)$ denote the *torsion subgroup* of $H_k(X; \mathbb{Z})$, that is, the set of elements of finite order of $H_k(X; \mathbb{Z})$. Then it is well-known from algebra that $H_k(X; \mathbb{Z}) \simeq \mathbb{Z}^{b_k(X)} \times T_k(X)$, where the rank $b_k(X)$ is called the kth *Betti number* of X. We already noted in the proof of Theorem 5.2 that the 0th Betti number $b_0(X)$ counts the number of connected components of X. For $k > 0$, $b_k(X)$ measures a more sophisticated "degree of connectivity".

In Bürgisser & Cucker (2003) it was shown that, for all $k \in \mathbb{N}$, the problem to compute the kth Betti number of the semilinear set given by a decision 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 $H_k(X;\mathbb{Z})$. 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 $\mathrm{T{\scriptsize ORSION}}_{\mathrm{add}}$ 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 5.3. *The problem* $\mathrm{T{\scriptsize ORSION}}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-hard under Turing reductions and belongs to* $\mathrm{EXP}_{\mathrm{add}}$.

Before giving the proof, we recall some facts from algebraic topology. We will write $I := [0, 1]$ for the closed unit interval.

The simplest space whose homology is not torsion free is the real projective plane. To describe it, recall that a *Moebius strip* M is the space obtained from I^2 by identifying the points $(0, t)$ with $(1, 1-t)$ on opposite edges in reverse orientation. The salient feature of M is that its boundary ∂M is homeomorphic to the circle S^1. It is well known (Massey 1977, Example 4.3, p. 9 or Stöcker & Zieschang 1988, Satz 1.4.18, p. 24) that when attaching to M a 2-cell by identifying the points on the cell's boundary S^1 with ∂M, one obtains the *real projective plane* $\mathbb{P}^2(\mathbb{R})$. Moreover, we have $H_1(\mathbb{P}^2(\mathbb{R});\mathbb{Z}) \simeq \mathbb{Z}_2$.

Let $\mathbb{P}^2(\mathbb{R}) \vee S^1$ be the space obtained from the disjoint union of $\mathbb{P}^2(\mathbb{R})$ and the circle S^1 by identifying a point of $\mathbb{P}^2(\mathbb{R})$ with one of S^1 (one-point union). Then

$$H_1(\mathbb{P}^2(\mathbb{R}) \vee S^1;\mathbb{Z}) \simeq \mathbb{Z}_2 \oplus \mathbb{Z}. \tag{5.4}$$

The following lemma provides the lower bound part of Theorem 5.3.

LEMMA 5.5. *The problem* $\mathrm{T{\scriptsize ORSION}}_{\mathrm{add}}$ *is* $\mathrm{PAR}_{\mathrm{add}}$*-hard under Turing reductions. The same holds when restricted to circuits with a fixed number k of input gates, for all $k \geq 5$.*

PROOF. In the proof of Theorem 5.2 we reduced an arbitrary language L in PSPACE to the problem $\mathrm{C{\scriptsize ONNECTED}}_{\mathrm{add}}$. In fact, we showed that from $x \in \{0,1\}^*$ one can compute in polynomial time a decision circuit describing a semilinear set $S'_n \subseteq \mathbb{R}^3$ such that $x \in L$ iff S'_n is connected. We will now modify this construction.

Recall that S'_n was obtained from the semilinear set $S_n \subseteq \mathbb{R}^3$ constructed in the proof of Lemma 5.1 by joining the images in $\mathbb{R}^3$ of the distinguished points

$(i(x), 0)$ and $(c_R, T+1)$ in G_n by a chain of line segments. Let $e = \{u, v\}$ be the first of these line segments.

In a first stage, we construct a twisted version of the cylinder $S'_n \times I \subseteq \mathbb{R}^4$ embedded in $\mathbb{R}^4$. For this we will use a set $\tau_e \subseteq \mathbb{R}^4$, which is obtained from $e \times I \subseteq \mathbb{R}^3$ essentially by twisting (in the fourth dimension) the opposite edges $\{u\} \times I$ and $\{v\} \times I$ by 180 degrees. We show now how to realize this with a semilinear set. For simplicity, we assume that e is the closed line segment connecting the origin $u = (0, 0, 0)$ and the point $(0, 1, 1)$. Then we define the *twist over* e as

$$\tau_e := \{0\} \times \big((I^2 \times \{0\}) \cup (\{0\} \times I^2) \cup (\{1\} \times I^2)\big),$$

see Figure 5.4 (where the 3-dimensional representation is accurate since the objects lie in $\{0\} \times \mathbb{R}^3 \simeq \mathbb{R}^3$ and points in $\mathbb{R}^4$ have coordinates (x, y, z, w)). The

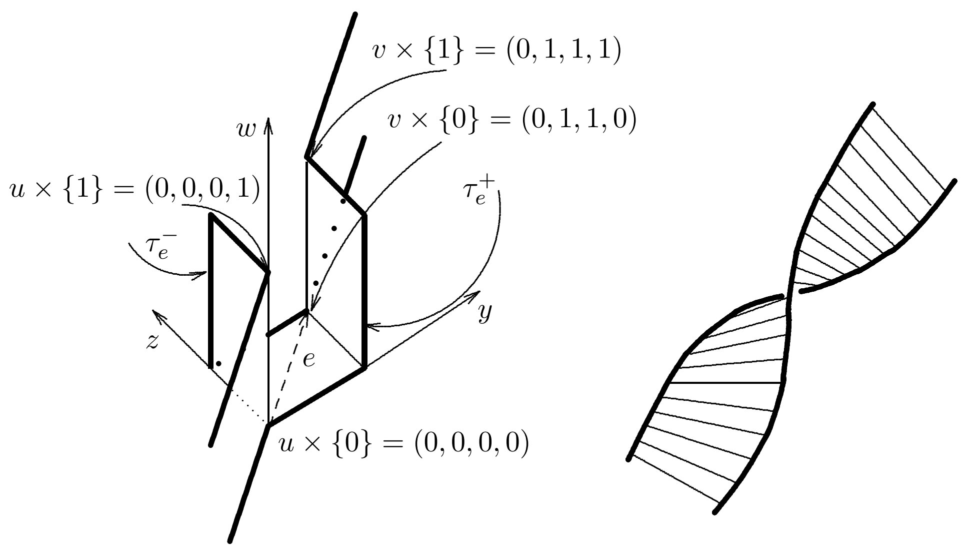

Figure 5.4: The twist τ_e and the portion of a Moebius strip it corresponds to.

chains of line segments τ_e^+ and τ_e^- drawn with thick lines in Figure 5.4 connect the point $u \times \{0\} = (0, 0, 0, 0)$ with $v \times \{1\} = (0, 1, 1, 1)$, and $u \times \{1\} = (0, 0, 0, 1)$ with $v \times \{0\} = (0, 1, 1, 0)$, respectively. Similarly, one can define τ_e for an arbitrary line segment e, but we refrain from explicitly doing so.

Consider now the following semilinear subset of $\mathbb{R}^4$

$$T_n := ((S'_n - e) \times I) \cup \tau_e$$

which is obtained from $S'_n \times I$ by replacing $e \times I$ with the twist τ_e. Moreover consider the set

$$B_n := ((S'_n - e) \times \{0,1\}) \cup \tau_e^+ \cup \tau_e^-$$

of "boundary lines" of T_n. Figure 5.5 and Figure 5.6 illustrate the sets S'_n, T_n, and B_n, the latter being drawn with thick lines. (Of course, these 3-dimensional pictures are not fully accurate since T_n is embedded in $\mathbb{R}^4$.)

Starting from the decision circuit describing the set S'_n, it is straightforward to design (and to compute in polynomial time) a decision circuit for T_n.

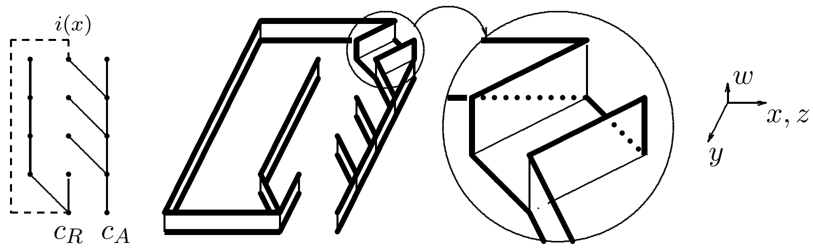

Figure 5.5: Illustration of S'_n, T_n, and B_n in the case where $x \in L$.

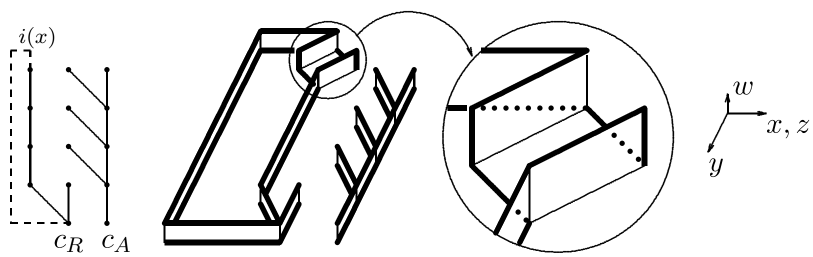

Figure 5.6: Illustration of S'_n, T_n, and B_n in the case where $x \notin L$.

In the case where $x \in L$, the set T_n is homeomorphic to $S'_n \times I$ and B_n corresponds to $S'_n \times \{0,1\}$ under this homeomorphism. Indeed, we can "untwist" the set in this case. By contracting the tree S'_n to a point we see that the pair (T_n, B_n) is homotopy equivalent to the pair $(I, \{0,1\})$.

In the case where $x \notin L$, let S_n^A denote the connected components of S'_n containing c_A. Then the set T_n is homeomorphic to the disjoint union of $S_n^A \times I$ with a space that can be continuously contracted to a Moebius strip M such that the boundary set is mapped into itself during the contraction. From this it follows that the pair (T_n, B_n) is homotopy equivalent to $(M \cup I, \partial M \cup \{0,1\})$ where $\cup$ denotes disjoint union.

We consider now the space $Q_n := T_n/B_n$ obtained by collapsing all points of B_n to a point. In the case where $x \in L$, we have ($\sim$ denoting homotopy equivalent spaces)

$$Q_n \sim I/\{0,1\} \sim S^1.$$

Since $H_0(S^1;\mathbb{Z}) \simeq H_1(S^1;\mathbb{Z}) \simeq \mathbb{Z}$ and all other homology groups of S^1 vanish, the homology of Q_n is torsion free.

To analyze the case where $x \notin L$, we claim that the space $M/\partial M$ obtained by collapsing the points in ∂M is homotopy equivalent to the real projective plane $\mathbb{P}^2(\mathbb{R})$. Indeed, note that $M/\partial M$ is homotopy equivalent to the space obtained from M by attaching a cone with base ∂M. The latter space is homeomorphic to the space obtained from M by attaching a 2-cell along ∂M and thus homeomorphic to $\mathbb{P}^2(\mathbb{R})$. Using this observation, it is easy to see that

$$Q_n \sim M/\partial M \vee S^1 \sim \mathbb{P}^2(\mathbb{R}) \vee S^1.$$

Hence in the case $x \notin L$ we have $H_1(Q_n;\mathbb{Z}) \simeq H_1(\mathbb{P}^2(\mathbb{R}) \vee S^1;\mathbb{Z}) \simeq \mathbb{Z}_2 \oplus \mathbb{Z}$ by (5.4) and thus the homology of Q_n is not torsion free.

It remains to realize Q_n, up to homotopy equivalence, as a semilinear set. This is achieved by the following semilinear set $\tilde{Q}$ in $\mathbb{R}^5$:

$$\tilde{Q} := (T_n \times \{0\}) \cup (B_n \times I) \cup (\mathbb{R}^4 \times \{1\}).$$

It is not hard to see that $\tilde{Q}$ is homotopy equivalent to Q_n. Moreover, a decision circuit for $\tilde{Q}$ can be easily computed from a decision circuit for T_n. $\square$

We finally provide the upper bound part of the proof of Theorem 5.3.

LEMMA 5.6. *The problem* $\text{TORSION}_{\text{add}}$ *is in* EXP_{add}.

PROOF. The proof is very similar to the one of Bürgisser & Cucker (2003, Proposition 5.22) in which the FPAR_{add}-upper bound for the computation of the Betti numbers was established. Therefore we only sketch the proof.

Assume we have group homomorphisms $\mathbb{Z}^m \xrightarrow{\alpha} \mathbb{Z}^n \xrightarrow{\beta} \mathbb{Z}^p$ such that $\beta \circ \alpha = 0$ and put $B := \operatorname{im}\alpha$, $Z := \ker\beta$, and $H := Z/B$. Then we have $H \simeq \mathbb{Z}^r \times \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_s\mathbb{Z}$, where $r \in \mathbb{N}$ is the rank and the *torsion coefficients* $d_1, \ldots, d_s$ are positive integers. The group H is torsion-free iff $s = 0$. It is well known that $d_1, \ldots, d_s$ can be obtained by computations of the Smith normal form of integer matrices, cf. Munkres (1984).

When studying the homology of a finite cell complex, the maps α and β are given by the incidence matrices between cells of contiguous dimensions. In the situation of a semilinear set $S_{\mathscr{C}}$ given by a circuit, there is a natural cell decomposition of $S_{\mathscr{C}}$ of exponential size, in which the cells and the incidence matrices are given in succinct representation. (A succinct representation of an integer matrix (a_{ij}) is a Boolean circuit computing a_{ij} from the index pair (i, j) given in binary.)

A crucial part of the proof of Bürgisser & Cucker (2003, Proposition 5.22) is the fact that the rank of an integer matrix can be computed in FNC. If there were a corresponding result for the computation of the Smith normal form of an integer matrix, then $\textsc{Torsion}_{\mathrm{add}} \in \mathrm{FPAR}_{\mathrm{add}}$ would follow along the lines of Bürgisser & Cucker (2003, Proposition 5.22). However, we cannot hope to provide an FNC-algorithm for computing the Smith normal form over $\mathbb{Z}$ unless progress is made on the question whether the gcd of integers can be computed in FNC, cf. Kaltofen *et al.* (1987). However, it has been known for a long time Kannan & Bachem (1979) that the Smith normal form over $\mathbb{Z}$ can be computed in FP. This results readily implies that the computation of the Smith normal form of an integer matrix given in succinct representation can be computed in FEXP. Using this, the assertion follows along the lines of Bürgisser & Cucker (2003, Proposition 5.22). □

6. Properties for the Zariski Topology

In this section we study the complexity of several topological properties of semilinear sets, where the topology considered is the Zariski topology.

6.1. Zariski Adherence, Denseness, and Closedness. The proof of the following result is analogous to the one of Proposition 4.1 and Proposition 4.3 and therefore omitted.

PROPOSITION 6.1. *The problem* $\textsc{ZClosed}_{\mathrm{add}}$ *is* $\mathrm{coNP}_{\mathrm{add}}$*-complete and the problem* $\textsc{ZAdh}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.* □

In Proposition 4.4 we showed that $\textsc{EDense}_{\mathrm{add}}$ is $\mathrm{coNP}_{\mathrm{add}}$-complete. By contrast, we prove the following.

PROPOSITION 6.2. *The problem* $\textsc{ZDense}_{\mathrm{add}}$ *is* $\mathrm{NP}_{\mathrm{add}}$*-complete.*

PROOF. Note that

$$\overline{S_{\mathscr{C}}}^{Z} = \overline{\bigcup_{\gamma \in \mathcal{A}_{\mathscr{C}}} D_\gamma}^{Z} = \bigcup_{\gamma \in \mathcal{A}_{\mathscr{C}}} \overline{D_\gamma}^{Z}. \tag{6.3}$$

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 $\mathrm{NP}_{\mathrm{add}}$ now follows from the fact that $\mathrm{DIM}_{\mathrm{add}}$ is in $\mathrm{NP}_{\mathrm{add}}$ (Bürgisser & Cucker 2003, Theorem 5.1).

The hardness follows from the reduction (4.2) in Proposition 4.1 which reduces $\mathrm{CBS}_{\mathrm{add}}$ to $\mathrm{ZDENSE}_{\mathrm{add}}$. □

6.2. Deciding Irreducibility. Irreducibility is a natural concept in algebraic geometry (Shafarevich 1974). For semilinear sets this notion can be defined as follows.

DEFINITION 6.4. *(i) A semilinear set $S \subseteq \mathbb{R}^n$ is called* Zariski-irreducible *if its Zariski closure is an affine space. We call a semilinear set* reducible *if it is not irreducible. (The empty set is considered to be irreducible.)*

(ii) The Zariski closure $\overline{S}^{Z}$ 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.*

Recall that the complexity class $\mathsf{P}^{\mathsf{NP}[\log]}$ 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. 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.

Both characterizations of $\mathsf{P}^{\mathsf{NP}[\log]}$ can be extended to the additive setting in the obvious way. Moreover, the proof of equivalence extends to the additive setting in a straightforward way (cf. Papadimitriou 1994, Theorem 17.7). We thus define:

DEFINITION 6.5. $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$ *is the class of problems decidable by a polynomial time additive machine which asks non adaptively a polynomial number of queries to problems in* $\mathrm{NP}_{\mathrm{add}}$.

The following is the main result of this section.

THEOREM 6.6. *The problem* $\mathrm{IRR}_{\mathrm{add}}$ *is* $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$*-complete.*

We first prove the upper bound.

Lemma 6.7. *The problem* $\mathrm{Irr}_{\mathrm{add}}$ *is in* $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$.

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 $d=\max\{k:$ (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 $\mathrm{Dim}_{\mathrm{add}}$ is known to be in $\mathrm{NP}_{\mathrm{add}}$ (Bürgisser & Cucker 2003), (i) is a query to a problem in $\mathrm{NP}_{\mathrm{add}}$. By Lemma 3.4, (ii) is a query to a problem in $\mathrm{coNP}_{\mathrm{add}}$. Since the queries are nonadaptive and the algorithm runs in polynomial time, the set $\mathrm{Irr}_{\mathrm{add}}$ is in $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$. □

The easy proof of the following technical lemma is left to the reader.

Lemma 6.8. **(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 finite nonempty union of reducible semilinear sets is reducible.* □

We turn now to the proof of the lower bound in Theorem 6.6.

Lemma 6.9. *The problem* $\mathrm{Irr}_{\mathrm{add}}$ *is* $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$*-hard under many-one reductions.*

Proof. Assume L is a problem in $\mathrm{P}_{\mathrm{add}}^{\mathrm{NP}_{\mathrm{add}}[\log]}$. Then we may assume that L is decided by a polynomial time additive machine asking non adaptively a polynomial number of queries to the $\mathrm{NP}_{\mathrm{add}}$-complete problem $\mathrm{ZDense}_{\mathrm{add}}$. 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 $\mathrm{ZDense}_{\mathrm{add}}$ 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 $\mathcal{E}_i^n$ of $\mathscr{C}_i^n$ on input x is an input to ZDENSE$_{\rm add}$. Thus $\mathcal{E}_i^n$ 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 $\mathcal{E}_i^n$ use the same number of selection gates $q(n) > 1$.) We denote by $\mathcal{A}_i$ the set of accepting paths of $\mathcal{E}_i^n$. Moreover, for $\gamma \in \mathcal{A}_i$, we denote by $D_{\gamma i} \subseteq S_i$ the corresponding leaf set, and write $\partial D_{\gamma i}$ for its Euclidean boundary.

The reduction (4.2) from the proof of Proposition 6.2 that reduces CBS$_{\rm add}$ to ZDENSE$_{\rm add}$ 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 $r(n) > 1$)

$$S_i \neq \emptyset \Longrightarrow \bigcup_{\gamma \in \mathcal{A}_i} \partial D_{\gamma i} \text{ is reducible.} \tag{6.10}$$

Our goal is to reduce L to IRR$_{\rm add}$. 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 6.8, 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:

$$\begin{aligned} s_i = 1 \iff \overline{S_i}^Z = \mathbb{R}^r \iff \overline{T_i}^Z = \mathbb{R}^r, \\ s_i = 0 \iff S_i = \emptyset \iff \overline{T_i}^Z = \{z\}. \end{aligned} \tag{6.11}$$

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:

$$x \in L \iff \Omega \text{ is irreducible.} \tag{6.12}$$

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

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

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

$$\Omega = \Omega_{\mathrm{I}} \quad \cup \bigcup_{\alpha \in \mathcal{Y}, j \in J(\alpha)} \big(\Omega^j_{\mathrm{II}}(\alpha) \cup \Omega_{\mathrm{III}}(\alpha)\big). \tag{6.13}$$

The following claim settles one direction of (6.12).

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 6.8(i) and (6.11). 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 (6.13), 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 (6.12). Indeed, assume $x \notin L$. Then $s \notin \mathcal{Y}$ and according to (6.13), Ω is a union of reducible sets and thus reducible.

It remains to prove Claim B. Let $\pi_j \colon \Pi \to \mathbb{R}^r, (u, y, a) \to y_j$ 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 (6.10) 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 right-hand inclusion is clear since $j \in J(\alpha)$ and thus $\alpha_j = 0$.

For the left-hand 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 $\mathsf{sg}(u) = \gamma$. We then have $(u, z, \ldots, z, y_j, z, \ldots, z, a) \in \Omega^j_{\mathrm{II}}(\alpha)$, 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. $\square$

6.3. Counting Irreducible Components.

THEOREM 6.14. *The problem* $\#\mathrm{IRR}_{\mathrm{add}}$ *is* $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$*-complete under Turing reductions.*

Proof. It follows from (6.3) that the irreducible components of $\overline{S_{\mathscr{C}}}^Z$ are some of the sets $\overline{D_\gamma}^Z$. Note, however, that some nonempty $\overline{D_\gamma}^Z$ may not be irreducible components since they are embedded in some higher dimensional components. Also, several paths γ may yield the same $\overline{D_\gamma}^Z$. A way to deal with these features is to associate to an irreducible component V of $\overline{S_{\mathscr{C}}}^Z$ the largest $\gamma \in \mathcal{A}_{\mathscr{C}}$ (say, with respect to lexicographical ordering) such that $V = \overline{D_\gamma}^Z$.

Let $\mathcal{I}_{\mathscr{C}} \subseteq \mathcal{A}_{\mathscr{C}}$ denote the set of paths associated to an irreducible component of $\overline{S_{\mathscr{C}}}^Z$ this way. Thus $|\mathcal{I}_{\mathscr{C}}|$ equals the number of irreducible components of $S_{\mathscr{C}}$.

The following algorithm decides membership of γ to $\mathcal{I}_{\mathscr{C}}$

input $(\mathscr{C}, \gamma)$
check that $\gamma \in \mathcal{A}_{\mathscr{C}}$ and $D_\gamma \neq \emptyset$, otherwise REJECT
check that for all $\gamma' \in \mathcal{A}_{\mathscr{C}}$ $(\overline{D_\gamma}^Z \subseteq \overline{D_{\gamma'}}^Z \Rightarrow \overline{D_\gamma}^Z = \overline{D_{\gamma'}}^Z)$ else REJECT
check that for all $\gamma' \in \mathcal{A}_{\mathscr{C}}$ $(\overline{D_\gamma}^Z = \overline{D_{\gamma'}}^Z \Rightarrow \gamma' \leq \gamma)$ else REJECT
ACCEPT

Note that by Lemma 3.4, the second line is decided in $\mathrm{P}_{\mathrm{add}}$ and the third and fourth line are in $\mathrm{coNP}_{\mathrm{add}}$. Therefore, the algorithm decides a set in $\mathrm{coNP}_{\mathrm{add}}$ and it follows that $\#\mathrm{Irr}_{\mathrm{add}} \in \mathrm{D}\# \cdot \mathrm{coNP}_{\mathrm{add}}$. By the additive version of Toda-Watanabe's Theorem (Bürgisser & Cucker 2003, Cor. 4.6), we have $\mathrm{D}\# \cdot \mathrm{coNP}_{\mathrm{add}} \subseteq \mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$, hence $\#\mathrm{Irr}_{\mathrm{add}} \in \mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$.

We now prove the hardness. Note first that the problem $\mathrm{CSat}_{\mathrm{add}}$ trivially belongs to $\mathrm{P}_{\mathrm{add}}^{\#\mathrm{Irr}_{\mathrm{add}}}$. Indeed, we have $S_{\mathscr{C}} = \emptyset$ iff the number of irreducible components of $S_{\mathscr{C}}$ is zero. Since $\mathrm{CSat}_{\mathrm{add}}$ is $\mathrm{NP}_{\mathrm{add}}$-complete, $\mathrm{NP}_{\mathrm{add}} \subseteq \mathrm{P}_{\mathrm{add}}^{\#\mathrm{Irr}_{\mathrm{add}}}$. Therefore, by Bürgisser & Cucker (2003, Theorem 5.1), we have $\mathrm{Dim}_{\mathrm{add}} \in \mathrm{P}_{\mathrm{add}}^{\#\mathrm{Irr}_{\mathrm{add}}}$.

It is now easy to design a Turing reduction from $\#\mathrm{CSat}_{\mathrm{add}}$ to $\#\mathrm{Irr}_{\mathrm{add}}$. On input a decision circuit $\mathscr{C}$, first decide whether $S_{\mathscr{C}}$ is finite using oracle calls to $\mathrm{Dim}_{\mathrm{add}}$, and hence to $\#\mathrm{Irr}_{\mathrm{add}}$. If $S_{\mathscr{C}}$ is not finite, return ∞, otherwise return the number of irreducible components of $S_{\mathscr{C}}$. Since $\#\mathrm{CSat}_{\mathrm{add}}$ is $\#\mathrm{P}_{\mathrm{add}}$-complete (Bürgisser & Cucker 2003, Theorem 3.6), this proves the hardness. □

Corollary 6.15. *For all $c, d \in \mathbb{N}$, $c > 0$, the problems $\#\mathrm{Irr}_{\mathrm{add}}^{(d)}$ and $\#\mathrm{Irr}_{\mathrm{add}}^{[c]}$ are $\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}$-complete under Turing reductions.*

Proof. The membership of both problems is shown similarly as in the proof of Theorem 6.14. We are going to prove the hardness of $\#\mathrm{Irr}_{\mathrm{add}}^{(d)}$ by reducing

#SAT to $\#\mathrm{IRR}^{(d)}_{\mathrm{add}}$. (The similar hardness proof for $\#\mathrm{IRR}^{[c]}_{\mathrm{add}}$ is left to the reader.)

For $\ell \le n-d$ and $s \in \{-1,1\}^n$ consider the affine function $L_{\ell,s} \colon \mathbb{R}^n \to \mathbb{R}$ defined by $L_{\ell,s}(x) = \sum_{j=\ell}^n s_j x_j - (n-\ell+1)$ and put $H_s := \{x \in \mathbb{R}^n \mid L_{1,s}(x) = 0, \ldots, L_{n-d,s}(x) = 0\}$. By construction, $s \in H_s$ and $\dim H_s = d$.

Assume that ϕ is a Boolean formula in conjunctive normal form with n variables. Consider a decision circuit $\mathscr{C}$ for the following algorithm (see Figure 6.1 that depicts the set accepted by the algorithm in the case $n = 2, d = 1$):

input $(x_1, \ldots, x_n)$
check that $x_i \neq 0$ for all $i \le n$ else REJECT
check that $x \in H_{s(x)}$ and $\|x_i - \mathsf{sg}(x_i)\|_\infty \le \frac{1}{2}$ else REJECT
check that $\mathsf{pos}(x_n) \in \{0,1\}^n$ satisfies ϕ else REJECT
ACCEPT

Clearly, $S_{\mathscr{C}}$ has the same number of irreducible components as ϕ has satisfying truth assignments. Moreover, all of these components have dimension d. This provides the desired reduction from #SAT to $\#\mathrm{IRR}^{(d)}_{\mathrm{add}}$. $\square$

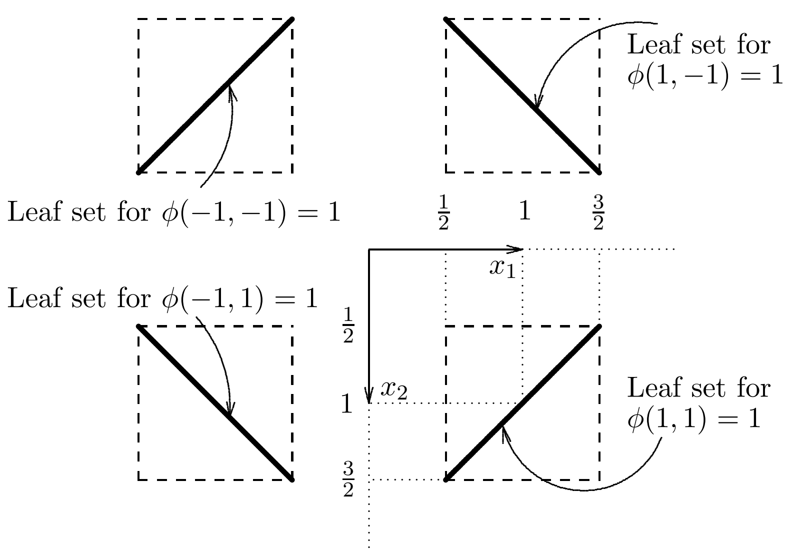

Figure 6.1: The leaf sets of $\mathscr{C}$ with $n = 2, d = 1$ where ϕ is a tautology.

COROLLARY 6.16. *For all $N \in \mathbb{N}$, $N > 0$, the problem $\#\mathrm{IRR}^{\{N\}}_{\mathrm{add}}$ is $\mathrm{FP}^{\#\mathrm{P}_{\mathrm{add}}}_{\mathrm{add}}$-complete under Turing reductions.*

Proof. The membership is immediate from Theorem 6.14. Without loss of generality we prove the hardness for $N = 1$, which is provided by the following reduction from #Sat to $\#\text{Irr}_{\text{add}}^{\{N\}}$. Assume that ϕ is a Boolean formula in conjunctive normal form with n variables. Consider a decision circuit $\mathscr{C}$ doing the following:

input $x \in \mathbb{R}$
check that $0 \le x < 2^n$ and $x \in \mathbb{N}$, otherwise REJECT
compute the sequence $\xi = (\xi_{n-1}, \ldots, \xi_0)$ of digits of x in binary
check that ξ is a satisfying assignment for ϕ, otherwise REJECT
ACCEPT

Note that the second line in the algorithm can be achieved in time polynomial in n by binary search. The third line can also be computed in time polynomial in n.

The non-empty leaf sets are irreducible, since they have dimension 0 and they are in one-to-one correspondence with the satisfying assignments of ϕ. $\square$

7. Completeness Results in the Turing Model

If an additive circuit has no constant gates (other than those with associated constants 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 we can consider, for instance, the following problem $\text{EAdh}_{\text{add}}^{\mathbb{Z}}$: given a constant-free additive circuit $\mathscr{C}$ with n input gates and a point $x \in \mathbb{Q}^n$, decide whether x belongs to the Euclidean closure of $S_{\mathscr{C}}$. Similarly, we can consider discrete versions of all the other problems dealt with in this paper and define $\text{Compact}_{\text{add}}^{\mathbb{Z}}$, $\text{Connected}_{\text{add}}^{\mathbb{Z}}$, $\#\text{Irr}_{\text{add}}^{\mathbb{Z}}$, etc.

We claim that all the completeness results shown in the previous two sections hold for the discrete versions of these problems with respect to the corresponding discrete complexity classes. A way to prove this is to carefully check all proofs given here. There is, however, an elegant way to get by free all the membership statements in these proofs.

Recall (see e.g., Bürgisser & Cucker 2003, Section 4), if $\mathcal{C}_{\text{add}}$ is a complexity class of decision problems for the additive model, its *constant-free Boolean part* $\text{BP}^0(\mathcal{C}_{\text{add}})$ is defined by

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

Here $\mathcal{C}_{\text{add}}^0$ is the subclass of $\mathcal{C}_{\text{add}}$ obtained by requiring all machines to be constant-free. That is, $\text{BP}^0(\mathcal{C}_{\text{add}})$ is the discrete complexity class obtained by restricting additive machines to be constant-free and inputs to these machines to be binary. It is known that (Blum *et al.* 1998; Bürgisser & Cucker 2003; Koiran 1994)

$$\mathrm{BP}^0(\mathrm{P}_{\mathrm{add}}) = \mathsf{P}, \quad \mathrm{BP}^0(\mathrm{NP}_{\mathrm{add}}) = \mathsf{NP}, \quad \mathrm{BP}^0(\mathrm{FP}_{\mathrm{add}}^{\#\mathrm{P}_{\mathrm{add}}}) = \mathsf{FP}^{\#\mathsf{P}}, \text{ etc.}$$

Now note that an immediate application of these equalities yields membership results. For instance, since $\mathrm{EA}{\small\mathrm{DH}}_{\mathrm{add}} \in \mathrm{NP}^0_{\mathrm{add}}$, we have $\mathrm{EA}{\small\mathrm{DH}}^{\mathbb{Z}}_{\mathrm{add}} \in \mathrm{BP}^0(\mathrm{NP}_{\mathrm{add}}) = \mathsf{NP}$.

The hardness results need to be proved in a different way. Again, we exemplify with $\mathrm{EA}{\small\mathrm{DH}}^{\mathbb{Z}}_{\mathrm{add}}$. First note that $\mathrm{CBS}^{\mathbb{Z}}_{\mathrm{add}}$ is NP-hard (a reduction from SAT to $\mathrm{CBS}^{\mathbb{Z}}_{\mathrm{add}}$ is immediate). Then, we note that the reduction from $\mathrm{CBS}_{\mathrm{add}}$ to $\mathrm{EA}{\small\mathrm{DH}}_{\mathrm{add}}$ given in Proposition 4.1 is constant-free (in the sense that it can be performed by a constant-free additive machine) and that it can be performed in polynomial time when restricted to binary inputs. In other words, when restricted to binary data it yields a reduction from $\mathrm{CBS}^{\mathbb{Z}}_{\mathrm{add}}$ to $\mathrm{EA}{\small\mathrm{DH}}^{\mathbb{Z}}_{\mathrm{add}}$ thus showing NP-hardness of $\mathrm{EA}{\small\mathrm{DH}}^{\mathbb{Z}}_{\mathrm{add}}$. Hardness for the other twenty-three problems is shown similarly.

We remark that for $\mathrm{REACH}^{\mathbb{Z}}_{\mathrm{add}}$, $\mathrm{CONNECTED}^{\mathbb{Z}}_{\mathrm{add}}$, and $\mathrm{TORSION}^{\mathbb{Z}}_{\mathrm{add}}$ we actually prove the PSPACE-hardness with respect to *many-one* reductions (instead of Turing reductions as in the additive model).

8. Final Remarks

Problems regarding the topology of a semilinear set other than those considered thus far easily come to mind. For instance,

$\mathrm{MANIFOLD}_{\mathrm{add}}$ (*Topological smoothness*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is a topological manifold.

$\mathrm{SIMPLYCONNECTED}_{\mathrm{add}}$ (*Simple Connectedness*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is simply connected.

$\mathrm{CONTRACTIBLE}_{\mathrm{add}}$ (*Contractibility*) Given a decision circuit $\mathscr{C}$, decide whether $S_{\mathscr{C}}$ is contractible.

The complexity of these problems is, as of today, an open problem. We have realized, however, that the discrete version of $\mathrm{SIMPLYCONNECTED}_{\mathrm{add}}$ is undecidable since the group triviality problem, which is undecidable (Adian 1957; Rabin 1958), reduces to it.

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 complete problems for $\mathsf{P}^{\mathsf{NP}[\log]}$. The first and third author would like to thank the Department of Mathematics and the Liu Bie Ju Centre for Mathematical Sciences of the City University of Hong Kong for the hospitality in April–June 2004, when most of the research on this paper was done. Peter Bürgisser was partially supported by DFG grant BU 1371 and the Paderborn Institute for Scientific Computation (PaSCo). Felipe Cucker and Paulin Jacobé de Naurois were partially supported by City University SRG grant 7001757.

References

S.I. Adian (1957). Unsolvability of certain algorithmic problems in the theory of groups (in Russian). *Trudy Moskov. Math. Obshch.* **6**, 231–298.

E.H. Bareiss (1968). Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation* **22**, 565–578.

L. Blum, F. Cucker, M. Shub & S. Smale (1998). *Complexity and Real Computation*. Springer-Verlag.

K.H. Borgwardt (1982). The average number of pivot steps required by the simplex method is polynomial. *Operations Research* **26**, 157–177.

P. Bürgisser & F. Cucker (2003). Counting Complexity Classes for Numeric Computations I: Semilinear Sets. *SIAM Journal on Computing* **33**, 227–260.

S. Buss & L. Hay (1991). On truth-table reducibility to SAT. *Information and Computation* **91**(1), 86–102.

A. Chandra, L. Stockmeyer & U. Vishkin (1984). Constant depth reducibility. *SIAM Journal on Computing* **13**, 423–439.

Dennis Cheung, Felipe Cucker & Yinyu Ye (2003). Linear programming and condition numbers under the real number computation model. In *Handbook of numerical analysis, Vol. XI*, Handb. Numer. Anal., XI, 141–207. North-Holland, Amsterdam.

F. Cucker & P. Koiran (1995). Computing over the reals with addition and order: Higher complexity classes. *Journal of Complexity* **11**, 358–376.

F. Cucker & M. Matamala (1996). On digital nondeterminism. *Mathematical Systems Theory* **29**, 635–647.

G.B. Dantzig & B.C. Eaves (1973). Fourier-Motzkin elimination and its dual. *Journal of Combinatorial Theory, Series A* **14**, 288–297.

T.K. Dey, H. Edelsbrunner & S. Guha (1999). Computational topology. In *Advances in discrete and computational geometry (South Hadley, MA, 1996)*, volume 223 of *Contemp. Math.*, 109–143. Amer. Math. Soc., Providence, RI.

J. Ferrante & C.W. Rackoff (1979). *The Computational Complexity of Logical Theories*. Number 718 in Lecture Notes in Computer Science. Springer-Verlag.

H. Fournier & P. Koiran (1998). Are lower bounds easier over the reals? In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing,* Dallas TX, 507–513.

H. Fournier & P. Koiran (2000). Lower bounds are not easier over the reals: Inside PH. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming ICALP 2000,* Geneva, Switzerland, number 1853 in Lecture Notes in Computer Science, 832–843.

A. Hatcher (2002). *Algebraic Topology.* Cambridge University Press, Cambridge.

F. Meyer auf der Heide (1984). A polynomial linear search algorithm for the n-dimensional Knapsack problem. *Journal of the ACM* **31**, 668–676.

L. Hemachandra (1989). The strong exponential hierarchy collapses. *Journal of Computer and System Sciences* **39**(3), 299–322.

E. Hemaspaandra, L.A. Hemaspaandra & J. Rothe (1997). Exact Analysis of Dodgson Elections: Lewis Carroll's 1876 Voting System is Complete for Parallel Access to NP. *Journal of the ACM* 806–825.

E. Kaltofen, M.S. Krishnamoorthy & B.D. Saunders (1987). Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM Journal on Algebraic and Discrete Methods* **4**, 683–690.

R. Kannan & A. Bachem (1979). Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing* **8**, 499–507.

L. Khachijan (1979). A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR* **244**, 1093–1096. (In Russian.) English translation in *Soviet Math. Dokl.* 20:191-194,1979.

P. Koiran (1994). Computing over the reals with addition and order. *Theoretical Computer Science* **133**, 35–47.

M. W. Krentel (1986). The complexity of optimization problems. In *Proceedings of the Eighteenth Annual ACM Symposium on the Theory of Computing,* Berkeley CA, 79–86.

W.S. Massey (1977). *Algebraic topology: an introduction.* Springer-Verlag, New York. Reprint of the 1967 edition, Graduate Texts in Mathematics, Vol. 56.

James R. Munkres (1984). *Elements of algebraic topology.* Addison-Wesley Publishing Company, Menlo Park, CA.

C.H. Papadimitriou (1994). *Computational Complexity.* Addison-Wesley.

C.H. Papadimitriou & S. Zachos (1983). Two remarks on the power of counting. In *Proc. 6th GI conference in Theoretical Computer Science*, number 145 in Lecture Notes in Computer Science, 269–276. Springer-Verlag.

M. Rabin (1958). Recursive unsolvability of group theoretic problems. *Annals of Mathematics* **67(2)**, 172–194.

M. Schechter (1998). Integration over a polyhedron: an application of the Fourier-Motzkin elimination method. *The American Mathematical Monthly* **105**, 246–251.

I.R. Shafarevich (1974). *Basic Algebraic Geometry.* Springer Verlag.

S. Smale (1983). On the average number of steps of the simplex method of linear programming. *Mathematical Programming* **27**, 241–262.

S. Smale (1998). Mathematical Problems for the Next Century. *The Mathematical Intelligencer* **20**(2), 7–15.

R. Stöcker & H. Zieschang (1988). *Algebraische Topologie.* Mathematische Leitfäden. B. G. Teubner, Stuttgart.

E. Tardos (1986). A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research* **34**, 250–256.

S. Toda (1991). PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing* **21**(2), 865–877.

S. Toda & O. Watanabe (1992). Polynomial time 1-Turing reductions from #PH to #P. *Theoretical Computer Science* **100**, 205–221.

L.G. Valiant (1979a). The complexity of computing the permanent. *Theoretical Computer Science* **8**, 189–201.

L.G. Valiant (1979b). The complexity of enumeration and reliability problems. *SIAM Journal on Computing* **8**, 410–421.

Manuscript received 3 March 2005, revised 12 March 2006

Peter Bürgisser
Department of Mathematics
University of Paderborn
D-33095 Paderborn
Germany
e-mail: `pbuerg@upb.de`

Felipe Cucker
Department of Mathematics
City University of Hong Kong
83 Tat Chee Avenue, Kowloon
Hong Kong
e-mail: `macucker@math.cityu.edu.hk`

Paulin Jacobé de Naurois
LORIA
615 rue du Jardin Botanique, BP 101
54602 Villers-lès-Nancy Cedex, Nancy
France
e-mail: `Paulin.De-Naurois@loria.fr`