Chicago Journal of Theoretical Computer Science

The MIT Press

Volume 1999, Article 8

Lower Bounds for Linear Satisfiability Problems

ISSN 1073–0486. MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1493 USA; (617)253-2889; *journals-ordersmit.edu, journals-infomit.edu.* Published one article at a time in LaTeX source form on the Internet. Pagination varies from copy to copy. For more information and other articles see:

- *http://mitpress.mit.edu/CJTCS/*
- *http://www.cs.uchicago.edu/publications/cjtcs/*
- *ftp://mitpress.mit.edu/pub/CJTCS*
- *ftp://cs.uchicago.edu/pub/publications/cjtcs*

The *Chicago Journal of Theoretical Computer Science* is abstracted or indexed in *Research Alert,®SciSearch, ®Current Contents®/Engineering Computing & Technology*, and *CompuMath Citation Index.®*
©1999 The Massachusetts Institute of Technology. Subscribers are licensed to use journal articles in a variety of ways, limited only as required to ensure fair attribution to authors and the journal, and to prohibit use in a competing commercial product. See the journal's World Wide Web site for further details. Address inquiries to the Subsidiary Rights Manager, MIT Press Journals; (617)253-2864; *journals-rightsmit.edu*.
The *Chicago Journal of Theoretical Computer Science* is a peer-reviewed scholarly journal in theoretical computer science. The journal is committed to providing a forum for significant results on theoretical aspects of all topics in computer science.

Editor-in-Chief: Janos Simon

Consulting Editors: Joseph Halpern, Stuart A. Kurtz, Raimund Seidel

Editors:

Martin Abadi	Greg Frederickson	John Mitchell
Pankaj Agarwal	Andrew Goldberg	Ketan Mulmuley
Eric Allender	Georg Gottlob	Gil Neiger
Tetsuo Asano	Vassos Hadzilacos	David Peleg
Laszló Babai	Juris Hartmanis	Andrew Pitts
Eric Bach	Maurice Herlihy	James Royer
Stephen Brookes	Ted Herman	Alan Selman
Jin-Yi Cai	Stephen Homer	Nir Shavit
Anne Condon	Neil Immerman	Eva Tardos
Cynthia Dwork	Howard Karloff	Sam Toueg
David Eppstein	Philip Klein	Moshe Vardi
Ronald Fagin	Phokion Kolaitis	Jennifer Welch
Lance Fortnow	Stephen Mahaney	Pierre Wolper
Steven Fortune	Michael Merritt	

Managing Editor: Michael J. O'Donnell

Electronic Mail: chicago-journalcs.uchicago.edu

Lower Bounds for Linear Satisfiability Problems

Jeff Erickson
(University of Illinois, Urbana-Champaign)
http://www.uiuc.edu/~jeffe
jeffe@uiuc.edu

6 August 1999

Abstract

We prove an $\Omega(n^{\lceil r/2 \rceil})$ lower bound for the following problem: For some fixed linear equation in r variables, given n real numbers, do any r of them satisfy the equation? Our lower bound holds in a restricted linear decision tree model, in which each decision is based on the sign of an arbitrary linear combination of r or fewer inputs. In this model, our lower bound is as large as possible. Previously, this lower bound was known only for a few special cases and only in more specialized models of computation.

Our lower bound follows from an adversary argument. We show that for any algorithm, there is a input that contains $\Omega(n^{\lceil r/2 \rceil})$ "critical" r-tuples, which have the following important property. None of the critical tuples satisfies the equation; however, if the algorithm does not directly test each critical tuple, then the adversary can modify the input, in a way that is undetectable to the algorithm, so that some untested tuple *does* satisfy the equation. A key step in the proof is the introduction of formal infinitesimals into the adversary input. A theorem of Tarski implies that if we can construct a single input containing infinitesimals that is hard for every algorithm, then for every decision tree algorithm there exists a corresponding real-valued input which is hard for that algorithm.

An extended abstract of this paper can be found in [Eri95].

1 Introduction

Many computational problems, especially in computational geometry, can be reduced to questions of the following form: Given n real numbers, do any r of them satisfy some fixed linear equation? More formally, let $\phi = \sum_{i=1}^{r} a_i t_i - b$ be a linear form with formal variables $t_1, t_2, \ldots, t_r$ and real coefficients $a_1, a_2, \ldots, a_r, b$, with $a_i \neq 0$ for all i. The *linear satisfiability problem* for ϕ is to determine, given n real numbers $x_1, x_2, \ldots, x_n$, whether there is a one-to-one map $\pi : \{1, 2, \ldots, r\} \hookrightarrow \{1, 2, \ldots, n\}$ such that

$$\phi(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(r)}) = \sum_{i=0}^{r} a_i x_{\pi(i)} - b = 0.$$

For almost all inputs (all but a measure-zero subset), there is no such map; consequently, if such a map does exist, we say that the input is *degenerate*. Any r-variable linear satisfiability problem can be solved in $O(n^{(r+1)/2})$ time when r is odd, or $O(n^{r/2} \log n)$ time when r is even. The algorithms that achieve these time bounds are quite simple (see Section 4); even so, these are the best known upper bounds.

The simplest example of a linear satisfiability problem is the well-known element uniqueness problem, which asks whether any two elements of a given multiset are equal; in this case, we have $\phi = t_1 - t_2$. Gajentaan and Overmars [GO95] describe a large class of "3SUM-hard" geometric problems,[1] each of which can be reduced, in subquadratic time, to deciding if a set of numbers has three elements whose sum is zero ($\phi = t_1 + t_2 + t_3$). Examples of 3SUM-hard problems include deciding whether a set of points in the plane contains three collinear points, whether a set of line segments can be split into two nonempty subsets by a line, whether a set of triangles has a simply connected union, or whether a line segment can be moved from one position and orientation to another in the presence of polygonal obstacles. Further examples are described in [ACH+96, dBdGO93, BvKT93, Mat95]. In a similar vein, Hernàndez Barrera [Her96] describes several problems that can be quickly reduced to the linear satisfiability problem for $\phi = t_1 + t_2 - t_3 - t_4$. Examples include computing the Minkowski sum of two polygons, sorting the vertices of a line arrangement, and determining whether one polygon can be translated to fit inside another. Higher-dimensional versions of several of these

[1] Some earlier papers use the more suggestive but potentially misleading term "n^2-hard"; see [BBG94].

problems can be reduced to a linear satisfiability problem with more variables. For example, deciding if a set of points in $\mathbb{R}^d$ contains $d+1$ points on a common hyperplane is at least as hard as the linear satisfiability problem for $\phi = t_1 + t_2 + \cdots + t_d$ [Eri99].

In this paper, we prove lower bounds on the complexity of linear satisfiability problems. We consider these problems under a restriction of the linear decision tree model of computation, called the *r-linear decision tree* model, in which each decision is based on the sign of an arbitrary affine combination of at most r elements of the input. Of particular interest are *direct queries*, which test the sign of the value of ϕ evaluated on r selected input elements. For example, for the element uniqueness problem, a direct query is a simple comparison. We show that any r-linear decision tree that solves an r-variable linear satisfiability problem must perform $\Omega(n^{\lceil r/2 \rceil})$ direct queries in the worst case. This matches the best known upper bounds when r is odd, and it is within a logarithmic factor when r is even. We also show, using results of Fredman [Fre76], that our lower bounds are as large as possible in the model we consider.

We prove our lower bounds using an adversary argument. Our approach is to derive, for each r-linear decision tree, a nondegenerate input with a large number of "critical" r-tuples that have the following important property: If an algorithm does not perform a direct query for every critical tuple, the adversary can modify the input, in a way that the algorithm cannot detect, so that some untested critical tuple lies in the zeroset of ϕ. Since the algorithm cannot distinguish between the original input and the modified input, even though the modified input is degenerate, the algorithm cannot produce the correct output for both inputs. It follows that any correct algorithm must check every critical tuple, so the number of critical tuples is a lower bound on the running time.

We use two new techniques to simplify our adversary construction. First, we allow our adversary inputs to contain formal infinitesimals instead of just real numbers. Tarski's transfer principle implies that if there is a hard input with infinitesimals, then for any algorithm, there is a corresponding real-valued input that is hard for that algorithm. Dietzfelbinger and Maass [DM88, Die89] presented a similar technique to prove lower bounds, using numbers that are "inaccessible" or have "different orders of magnitude"; an early version of this technique was also used by Fredman [Fre76]. Unlike their technique, utilizing infinitesimals makes it possible, and indeed sufficient, to derive a single adversary input for any problem, rather than explicitly

constructing a different input for every algorithm. Infinitesimals have been used extensively in geometric perturbation techniques [EM90, EC91, Yap90], in algorithms dealing with real semialgebraic sets [Can88, Can93], and in other lower-bound arguments [GKMS97, GV96].

Second, we allow our adversary inputs to be degenerate. That is, both the original adversary input and the modified input contain r-tuples in the zeroset of ϕ. Although it appears that such inputs cannot be used in an adversary argument, since the adversary's modification does not change the correct output, we show that a degenerate adversary input can always be perturbed slightly, resulting in a new nondegenerate adversary input with just as many critical tuples as the original.

1.1 Previous results

For any constant $r \geq 2$, an $\Omega(n \log n)$ lower bound for any r-variable linear satisfiability problem follows from the techniques of Dobkin and Lipton [DL79] in the (unrestricted) linear decision tree model. They observed that the set of inputs following a fixed computational path through a linear decision tree is connected. Since the set of nondegenerate inputs has $n^{\Omega(n)}$ connected components, even when $r = 2$, any linear decision tree must have $n^{\Omega(n)}$ leaves and, therefore, must have depth $\Omega(n \log n)$.

Dobkin and Lipton's techniques were generalized to higher-degree algebraic decision trees by Steele and Yao [SY82] and to algebraic computation trees by Ben-Or [Ben83], and the $\Omega(n \log n)$ lower bound holds in these models as well. Several more advanced techniques have been developed for proving lower bounds in these models (see, for example, [BLY92, GKMS97, GKV97, Yao95, Yao97]), but none of them improve the $\Omega(n \log n)$ lower bound.

Fredman [Fre76] proved an $\Omega(n^2)$ lower bound on the number of simple comparisons required, given two sets X and Y, to sort the elements of the multiset $X+Y = \{x+y \mid x \in X, y \in Y\}$ or to decide whether it contains any duplicate elements. "Sorting $X + Y$" is closely related to the four-variable linear satisfiability problem with $\phi = t_1+t_2-t_3-t_4$, and our results generalize Fredman's lower bound to arbitrary 4-linear decision trees.

Fredman's result was generalized by Dietzfelbinger [Die89], who derived an $\Omega(n^{r/2})$ lower bound on the depth of any comparison tree that determines, given a set of n reals, whether any two disjoint subsets of size $r/2$ have the same sum. In our terminology, he proves a lower bound for the specific

r-variable linear satisfiability problem with

$$\phi = \sum_{i=1}^{r/2} t_i - \sum_{i=1}^{r/2} t_{i+r/2},$$

in a model that allows only direct queries, for all even r. Dietzfelbinger claimed that his lower bound also holds in a more general restriction of the r-linear decision tree model, where every query polynomial has at most $r/2$ positive coefficients and at most $r/2$ negative coefficients. Our lower bound generalizes Dietzfelbinger's lower bound to arbitrary r-linear decision trees.

More recent techniques of Erickson and Seidel [ES95] (but see also [ES97]) and Erickson [Eri99] can be used to prove lower bounds for certain linear satisfiability problems, in a model of computation that allows only direct queries. However, there are still several such problems for which these techniques appear to be inadequate.

Our lower bounds should be compared with the following result of Meyer auf der Heide [Mey84]: For any fixed n, there exists a linear decision tree of depth $O(n^4 \log n)$ that solves the n-dimensional knapsack problem, "Given a set of n real numbers, does any subset sum to 1?" His nonuniform algorithm can be adapted to solve any of the linear satisfiability problems we consider, in the same amount of time [DM88]. Thus, there is no hope of proving lower bounds bigger than $\Omega(n^4 \log n)$ for any linear satisfiability problem in the full linear decision tree model. We reiterate that our lower bounds apply only to linear decision trees where the number of terms in any query is bounded by a constant.

Seidel [Sei97] has recently shown that, given three sets A, B, C, each containing n integers between 0 and m, we can determine whether there are elements $a \in A, b \in B, c \in C$ such that $a + b = c$, in time $O(n + m \log m)$. (Note that this problem is 3SUM-hard [GO95]!) Seidel's algorithm transforms the sets A, B, C into bit vectors, computes the integer vector representing the multiset $A+B$ using a fast Fourier transform, and compares it to the bit vector of C. This algorithm can be modified to solve any r-variable linear satisfiability problem, where the input consists of r sets of n bounded integers and the coefficients of ϕ are integers, in time $O(n + (r \log r)(m \log m))$. However, Seidel's algorithm cannot be modeled even as an algebraic decision tree. We discuss the bit length of our adversary inputs in Section 3.4.

1.2 Outline

The rest of the paper is organized as follows. Section 2 provides some useful definitions, including a formal definition of our model of computation. We prove our main theorem in Section 3. In Section 4, we establish matching nonuniform upper bounds. Finally, in Section 5, we offer our conclusions and suggest directions for further research.

2 Background and definitions

2.1 Hyperplane arrangements

A *hyperplane* h in $\mathbb{R}^n$ is an $(n-1)$-dimensional affine subspace, that is, a set of the form

$$h = \left\{ (x_1, x_2, \ldots, x_n) \;\middle|\; \sum_{i=0}^{n} \alpha_i x_i = \beta \right\}$$

or, more simply, $h = \{X \mid \langle X, \alpha \rangle = \beta\}$, for some real coefficients $\alpha_1, \ldots, \alpha_n, \beta$, where at least one α_i is not zero. The complement of a hyperplane h consists of a *positive halfspace* $h^+ = \{X \mid \langle X, \alpha \rangle > \beta\}$ and a *negative halfspace* $h^- = \{X \mid \langle X, \alpha \rangle < \beta\}$.

Any finite set $H = \{h_1, h_2, \ldots, h_N\}$ of hyperplanes in $\mathbb{R}^n$ defines a cell complex, called an *arrangement*. Each cell is a maximal connected subset of $\mathbb{R}^n$ contained in the intersection of a fixed subset of H and disjoint from any other hyperplane in H. The *dimension* of a cell is the dimension of the smallest affine subspace that contains it. For example, the n-dimensional cells are the connected components of $\mathbb{R}^n \setminus \bigcup_{i=1}^{N} h_i$. An arrangement of N hyperplanes in $\mathbb{R}^n$ has $O(N^n)$ cells [Ede87].

The closure of every cell in a hyperplane arrangement is a *(convex) polyhedron*. More generally, a polyhedron is the intersection of finite number of hyperplanes and closed halfspaces. A bounded polyhedron is called a *polytope*. A hyperplane *supports* a polyhedron if it intersects the polyhedron but does not intersect its relative interior. The intersection of a polyhedron and one of its supporting hyperplane is a *face* of the polyhedron; faces are themselves lower-dimensional polyhedra. A $(k-1)$-dimensional face of a k-dimensional polyhedron is called a *facet*, and a $(k-2)$-dimensional face is called a *ridge*. Each ridge is contained in exactly two facets. For example, a (three-dimensional) cube has 6 facets, 12 ridges, and a total of 26 faces.

We refer the reader to Edelsbrunner's monograph [Ede87] for further details on hyperplane arrangements and to Ziegler's lecture notes [Zie94] for a thorough introduction to the theory of convex polytopes and polyhedra.

2.2 r-linear decision trees

We now formally define our model of computation. A *linear decision tree* is a ternary tree in which each interior node v is labeled with a *query polynomial* $q_v \in \mathbb{R}[t_1, \ldots, t_n]$ of degree 1, and the outgoing edges of each interior node are labeled -1, 0, and $+1$. Each leaf is labeled with some value; for our purposes, these values are all either "yes" or "no." Given an input $X \in \mathbb{R}^n$, we compute with such a tree by traversing a path from the root to a leaf. At each node v on this path, we evaluate the sign of $q_v(X)$ and then proceed recursively in the appropriate subtree. When we reach a leaf, we return its label as the output of the algorithm. This definition is essentially the same as that given by Steele and Yao [SY82]; linear decision trees are also equivalent to the "linear search algorithms" investigated by Meyer auf der Heide [Mey84]. An *r-linear* decision tree is a linear decision tree, each of whose query polynomials has at most r linear terms (and possibly a constant term).

Linear decision trees have a natural geometric interpretation. Each query polynomial induces a hyperplane in the space $\mathbb{R}^n$ of possible inputs. At each internal node of the tree, we branch according to whether the input point X is on the corresponding query hyperplane (0), in its positive halfspace ($+1$), or in its negative halfspace (-1). If H_A is the set of hyperplanes induced by the query polynomials in a linear decision tree A, all the inputs in the same cell in the arrangement of H_A traverse the same root-to-leaf path in A. In other words, A cannot distinguish between two inputs in the same cell. In an r-linear decision tree, every query hyperplane is parallel to all but r of the coordinate axes.

An r-variable linear satisfiability problem asks whether a given point in $\mathbb{R}^n$ lies on a fixed set H_ϕ of $\Theta(n^r)$ hyperplanes, each parallel to all but r of the coordinate axes. Our main result can be stated geometrically as follows. There is an n-dimensional cell C in the arrangement of H_ϕ with $\Omega(n^{\lceil r/2 \rceil})$ boundary facets. Given a point $X \in C$ as input, if an algorithm fails to check whether X lies "inside" each boundary facet of C, the adversary can undetectably move X onto some unchecked boundary facet and, thus, onto a hyperplane in H_ϕ.

2.3 Ordered fields and infinitesimals

An *ordered field* is a field with a strict linear ordering $<$ compatible with the field operations. A *real closed field* is an ordered field, no proper algebraic extension of which is also an ordered field. The *real closure* $\widetilde{K}$ of an ordered field K is the smallest real closed field that contains it. We refer the interested reader to [BCR87, HRR91] for further details and more formal definitions, and to [Can88, Can93] for previous algorithmic applications of real closed fields.

A formula in the first-order theory of the reals, or more simply, a *first-order formula*, is a quantified Boolean combination of polynomial equations and inequalities. An *elementary formula* is a first-order formula with no free variables, in which every polynomial has real coefficients. An elementary formula *holds in* an ordered field K if and only if the formula is true when the range of the quantifiers is the field K and addition, multiplication, and comparisons are interpreted as ordered field operations in K. Obviously, this makes sense only if K contains the coefficients of the formula; every ordered field we consider is an extension field of the reals.

The following principle was originally proven by Tarski [Tar51] in a slightly different form. See [BCR87] for a proof of this version.

The transfer principle *Let $\widetilde{K}$ be a real closed extension field of $\mathbb{R}$. An elementary formula holds in $\widetilde{K}$ if and only if it holds in $\mathbb{R}$.*

For any ordered field K, we let $K(\varepsilon)$ denote the ordered field of rational functions in ε with coefficients in K, where ε is positive but less than every positive element of K. In this case, we say that ε is *infinitesimal in* K. We use towers of such field extensions. In such an extension, the order of the infinitesimals is specified by the description of the field. For example, in the ordered field $\mathbb{R}(\varepsilon_1, \varepsilon_2, \varepsilon_3) = \mathbb{R}(\varepsilon_1)(\varepsilon_2)(\varepsilon_3)$, ε_1 is infinitesimal in the reals, ε_2 is infinitesimal in $\mathbb{R}(\varepsilon_1)$, and ε_3 is infinitesimal in $\mathbb{R}(\varepsilon_1, \varepsilon_2)$.

An important property of such a field (in fact, the only property we really need) is that the sign of any element $a_0 + a_1\varepsilon_1 + a_2\varepsilon_2 + a_3\varepsilon_3 \in \mathbb{R}(\varepsilon_1, \varepsilon_2, \varepsilon_3)$, where each of the coefficients a_i is real, is given by the sign of the first nonzero coefficient; in particular, the element is zero if and only if every a_i is zero. In other words, we can treat the set of elements of the form $a_0+a_1\varepsilon_1+a_2\varepsilon_2+a_3\varepsilon_3$ as the vector space $\mathbb{R}^4$ with a lexicographic ordering. Our constructions never use higher-order elements such as $\varepsilon_1^2 + \varepsilon_2/\varepsilon_3$.

Let K be any ordered field extension of the reals. Since K is ordered, and since any real polynomial can be thought of as a function from K to K, it is reasonable to talk about the behavior of any linear decision tree given elements of K as input. We emphasize that query polynomials always have real coefficients, even when we consider more general inputs.

3 The main theorem

In this section, we prove the following theorem.

Theorem 1 *Any r-linear decision tree that solves an r-variable linear satisfiability problem must have depth $\Omega(n^{\lceil r/2 \rceil})$.*

Throughout this section, let $\phi = \sum_{i=1}^{r} a_i t_i - b$ denote a fixed linear form with formal variables $t_1, t_2, \ldots, t_r$ and real coefficients $a_1, \ldots, a_r, b$, where $a_i \neq 0$ for all i. We call the ordered r-tuple $(x_{\pi(1)}, \ldots, x_{\pi(r)})$ a *satisfying tuple* if it lies in the zeroset of ϕ, that is, if $\phi(x_{\pi(1)}, \ldots, x_{\pi(r)}) = 0$. We say that a set X is *degenerate* if it contains the elements of a satisfying tuple. For any set X, we call an ordered r-tuple of elements of X *critical* if the following properties are satisfied:

(1) The tuple is not in the zeroset of ϕ.

(2) There exists another *collapsed* set $\hat{X}$, such that the corresponding tuple in $\hat{X}$ is in the zeroset of ϕ but the sign of every other real linear combination of r or fewer elements is the same for both sets.

In other words, the only way for an r-linear decision tree to distinguish between X and $\hat{X}$ is to perform a direct query on the critical tuple. Critical tuples play the same role in our adversary argument as Dietzfelbinger's "fooling pairs" [Die89] and Erickson and Seidel's "collapsible simplices" [ES95, Eri99].

To prove our lower bound, it would suffice to prove the existence of a nondegenerate input X with several critical tuples. If an r-linear decision tree algorithm did not perform a direct query for each critical tuple, given X as input, then an adversary could "collapse" one of the untested tuples. The algorithm would be unable to distinguish between the original input X and the modified input $\hat{X}$, even though one would be degenerate and the other would not. Thus, the number of critical tuples would be a lower bound on the running time of any algorithm.

Unfortunately, this approach seems to be doomed from the start. For any two sets X and X' of real numbers, there are an infinite number of query polynomials that are positive at X and negative at X'. It follows that critical tuples are impossible. Moreover, no single input is hard for every algorithm, since for any set X of n real numbers, there is an algorithm that requires only n queries to decide whether X is degenerate.

To avoid these problems, we allow our adversary inputs to contain elements of an ordered extension field of the form $\mathbb{R}(\varepsilon_1, \ldots, \varepsilon_m)$. Allowing the adversary to use infinitesimals lets us construct a set with several critical tuples (Lemma 2), even though such sets are impossible if we restrict ourselves to real-valued inputs.

The algorithms we consider are required to behave correctly only when they are given real input. Therefore, the infinitesimal inputs we construct cannot be used directly in our adversary argument. The second step in our proof (Lemma 3) is to derive, for each r-linear decision tree, a corresponding real-valued input with several *relatively* critical tuples (defined below). This step follows from our infinitesimal construction by a straightforward application of Tarski's transfer principle. We emphasize that for each algorithm, we obtain a different real-valued input.

Finally, the adversary inputs we construct in the first step (and by implication, the real inputs we get by invoking the transfer principle) contain several satisfying r-tuples. Thus, the critical tuples do not immediately imply a lower bound, since both the original input and any collapsed input are degenerate. In the final step of the proof (Lemma 4), we use simple properties of hyperplane arrangements and convex polyhedra to show that these degenerate inputs can be perturbed slightly, resulting in nondegenerate inputs with the same critical tuples. Thus, for each r-linear decision tree, we obtain a corresponding nondegenerate input with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples. Our lower bound then follows by the previous adversary argument.

3.1 The infinitesimal adversary input

Our construction relies on the existence of an integer matrix with two special properties.

Lemma 1 *There exists an $r \times \lfloor r/2 \rfloor$ integer matrix M satisfying the following conditions:*

(1) There are $\Omega(n^{\lceil r/2 \rceil})$ vectors $v \in \{1, 2, \ldots, n\}^r$ such that $M^\top v = 0$.

(2) Every set of $\lfloor r/2 \rfloor$ rows of M forms a nonsingular matrix.

Proof Consider the matrix M with entries

$$m_{ij} = \begin{cases} i^{j-1} & \text{if } 1 \le i \le \lceil r/2 \rceil, \\ -1 & \text{if } i = j + \lceil r/2 \rceil, \\ 0 & \text{otherwise,} \end{cases}$$

where $1 \le i \le r$ and $1 \le j \le \lceil r/2 \rceil$. The first $\lceil r/2 \rceil$ rows of M form a rectangular Vandermonde matrix, and the last $\lfloor r/2 \rfloor$ rows form a negative identity matrix. We claim that this matrix satisfies the conditions of the lemma.

We can choose a vector $v = (v_1, v_2, \ldots, v_r) \in \{1, 2, \ldots, n\}^r$ such that $M^\top v = 0$ as follows. Let $m_{\max} = \lceil r/2 \rceil^{\lceil r/2 \rceil - 1}$ denote the largest element of M. For each $1 \le i \le \lceil r/2 \rceil$, choose v_i arbitrarily in the range

$$1 \le v_i \le \left\lfloor \frac{n}{\lceil r/2 \rceil m_{\max}} \right\rfloor,$$

and for all $1 \le j \le \lfloor r/2 \rfloor$, let

$$v_{j+\lceil r/2 \rceil} = \sum_{i=1}^{\lceil r/2 \rceil} m_{ij} v_i = \sum_{i=1}^{\lceil r/2 \rceil} i^j v_i.$$

Each $v_{j+\lceil r/2 \rceil}$ is a positive integer in the range $\lceil r/2 \rceil \le v_j \le n$. We easily verify that $M^\top v = 0$. There are

$$\left\lfloor \frac{n}{\lceil r/2 \rceil m_{\max}} \right\rfloor^{\lceil r/2 \rceil} = \left\lfloor \frac{n}{\lceil r/2 \rceil^{\lceil r/2 \rceil}} \right\rfloor^{\lceil r/2 \rceil} = \Omega\left(n^{\lceil r/2 \rceil}\right)$$

different ways to choose the vector v. Thus, M satisfies condition (1).

Let M' be a matrix consisting of $\lfloor r/2 \rfloor$ arbitrary rows of M. We can write

$$M' = P \begin{pmatrix} V & W \\ 0 & -I \end{pmatrix},$$

where P is an $\lfloor r/2 \rfloor \times \lfloor r/2 \rfloor$ permutation matrix, V is a (possibly empty) square minor of a Vandermonde matrix, and I is a (possibly empty) identity matrix. (W is also a minor of a Vandermonde matrix, but this is unimportant.) Since P, V, and I are all nonsingular, so is M'. Thus, M satisfies condition (2). □

Lemma 2 *There exists a set $X \in K^n$ with $\Omega(n^{\lceil r/2 \rceil})$ critical tuples, for some ordered field K.*

Proof We construct such a set $X \in K^n$, where

$$K = \mathbb{R}(\Delta_1, \ldots, \Delta_{r-1}, \delta_1, \ldots, \delta_{\lfloor r/2 \rfloor}, \varepsilon).$$

We assume without loss of generality that n is a multiple of r.

Let $M = (m_{ij})$ be the matrix given by Lemma 1. Our set X is the union of r smaller sets X_i, each containing n/r elements x_{ij} defined as

$$x_{ij} = \frac{1}{a_i} \left(\frac{b}{r} + (-1)^i (\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik} j \delta_k + j^2 \varepsilon \right)$$

for all $1 \le i \le r$ and $1 \le j \le n/r$. For notational convenience, we define $\Delta_0 = \Delta_r = 0$.

We claim that any tuple $(x_{1p_1}, \ldots, x_{rp_r})$, where the indices p_i satisfy the matrix equation $M^\top(p_1, \ldots, p_r) = 0$, is critical. By condition (1) of Lemma 1, there are $\Omega((n/r)^{\lceil r/2 \rceil}) = \Omega(n^{\lceil r/2 \rceil})$ such tuples. The corresponding collapsed input $\hat{X}$ has elements

$$\hat{x}_{ij} = \frac{1}{a_i} \left(\frac{b}{r} + (-1)^i (\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik} j \delta_k + (j - p_i)^2 \varepsilon \right)$$

or, more succinctly, $\hat{x}_{ij} = x_{ij} + (p_i^2 - 2jp_i)\varepsilon/a_i$.

For example, in the simplest nontrivial case $r = 3$, the set X has elements in the field $\mathbb{R}(\Delta_1, \Delta_2, \delta_1, \varepsilon)$. If we take $M = (1, 1, -1)^\top$, then X contains the following elements, where $1 \le j \le n/3$:

$$\begin{aligned}
x_{1j} &= (b/3 - \Delta_1 \qquad\quad + j\delta_1 + j^2\varepsilon)/a_1, \\
x_{2j} &= (b/3 + \Delta_1 + \Delta_2 + j\delta_1 + j^2\varepsilon)/a_2, \\
x_{3j} &= (b/3 \qquad - \Delta_2 - j\delta_1 + j^2\varepsilon)/a_3.
\end{aligned}$$

The indices of each allegedly critical tuple satisfy the equation $p_1 + p_2 = p_3$, and the elements of the corresponding collapsed input $\hat{X}$ are

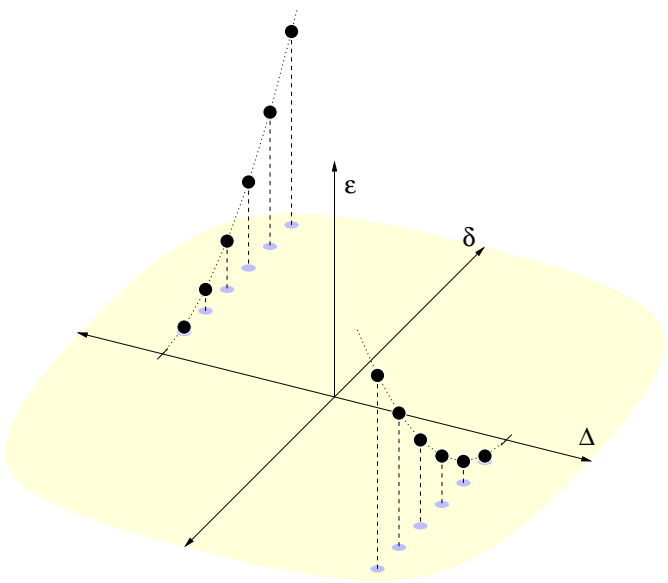
$$\begin{aligned}
\hat{x}_{1j} &= (b/3 - \Delta_1 \qquad\quad + j\delta_1 + (j - p_1)^2\varepsilon)/a_1, \\
\hat{x}_{2j} &= (b/3 + \Delta_1 + \Delta_2 + j\delta_1 + (j - p_2)^2\varepsilon)/a_2, \\
\hat{x}_{3j} &= (b/3 \qquad - \Delta_2 - j\delta_1 + (j - p_3)^2\varepsilon)/a_3.
\end{aligned}$$

Before continuing the proof, let us attempt to provide some intuition about our construction; the same intuition can be applied to Dietzfelbinger's construction [Die89]. To keep things simple, consider the case where $a_i = 1$ for all i and $b = 0$. We can write any linear expression $\sum_{i=1}^{r} \alpha_i x_{i\pi(i)}$, where $\alpha_i \in \mathbb{R}$, as a real linear combination of the infinitesimals $\Delta_i, \delta_k, \varepsilon$. The sign of such an linear combination is determined by the sign of the most significant nonzero coefficient. Each "level" of infinitesimals restricts which expressions of this form can possibly equal zero. Specifically, the Δ terms ensure that each coefficient $\alpha_i = 1$, and the δ terms ensure that the indices $\pi(i)$ satisfy the equation $M^\top(\pi(1), \ldots, \pi(r)) = 0$. The remaining expressions are direct queries on (allegedly) critical tuples; these are the only expressions whose signs can be changed by the adversary. The ε terms ensure that *no* such expression is equal to zero and that a corresponding expression in $\hat{X}$ equals zero if and only if $\pi(i) = p_i$ for all i.

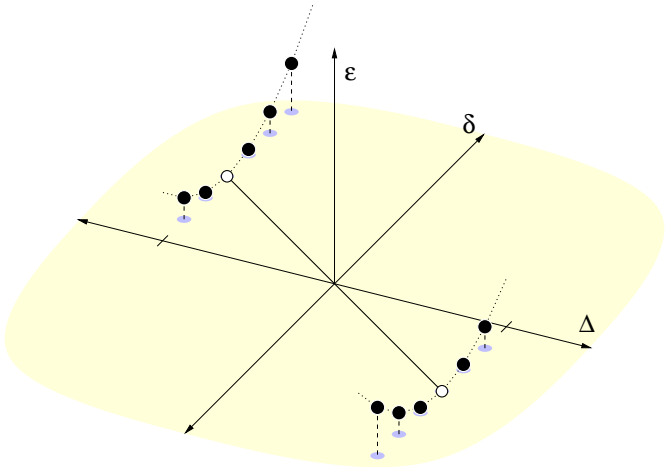
We can interpret any real linear combination of infinitesimals in K geometrically as an element of the lexicographically ordered vector space $\mathbb{R}^{r+\lfloor r/2 \rfloor}$. Figure 1 illustrates our adversary construction for the case $\phi = t_1 + t_2$ as a set of points or vectors in $\mathbb{R}^3$. The adversary set X contains the points $(-1, i, i^2), (1, -1, i^2)$ for all $1 \le i \le n/2$. For each $1 \le p \le n/2$, we have a collapsed set $\hat{X}$, which contains the points $(-1, i, (i-p)^2), (1, -i, (i-p)^2)$. "Collapsing" a critical pair involves "rolling" the two parabolas containing the points so that the desired pair lies in the (Δ, δ) plane. Clearly, every pair of points in X is linearly independent. Exactly one pair of points in $\hat{X}$ is linearly dependent, and those two points sum to zero. Otherwise, every linear combination of two points in X has the same sign as the corresponding points in $\hat{X}$.

We now continue our proof of Lemma 2. Fix an r-tuple $(x_{1p_1}, \ldots, x_{rp_r})$ such that $M^\top(p_1, \ldots, p_r) = 0$. We verify that $\phi(x_{1p_1}, \ldots, x_{rp_r}) \neq 0$ as follows.

$$\begin{aligned}
\phi(x_{1p_i}, \ldots, x_{rp_r}) &= \sum_{i=1}^{r} \left(\frac{b}{r} + (-1)^i(\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik} p_i \delta_k + p_i^2 \varepsilon \right) - b \\
&= \sum_{i=1}^{r} (-1)^i(\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} \left(\sum_{i=1}^{r} m_{ik} p_i \right) \delta_k + \sum_{i=1}^{r} p_i^2 \varepsilon \\
&= \sum_{i=1}^{r} p_i^2 \varepsilon > 0.
\end{aligned}$$



(a)



(b)

Figure 1: (a) The original adversary set X for the case $\phi = t_1 + t_2$. (b) A collapsed set $\hat{X}$; the white points lie in the (Δ, δ) plane.

Similar calculations show that $\phi(\hat{x}_{1p_i}, \ldots, \hat{x}_{rp_r}) = 0$.

To show that the tuple is critical, it remains to show that every other r-linear query has the same sign when evaluated at both X and $\hat{X}$. To distinguish between the query polynomials and their value at a particular input, let t_{ij} be the formal variable corresponding to each element x_{ij} in the set X above.

Consider the query polynomial $Q = \sum_{i=1}^{r} Q_i - \beta$, where for each i,

$$Q_i = a_i \sum_{j=1}^{n/r} \alpha_{ij} t_{ij},$$

and at most r of the real coefficients α_{ij} are not zero. We refer to t_{ij} as a *query variable* if its coefficient α_{ij} is not zero. For notational convenience, define

$$A_i = \sum_{j=1}^{n/r} \alpha_{ij} \qquad \text{and} \qquad J_i = \sum_{j=1}^{n/r} \alpha_{ij} j$$

for each i and

$$B = \frac{b}{r} \sum_{i=1}^{r} A_i - \beta.$$

We can rewrite the query expression $Q(X)$ as a real linear combination of the infinitesimals as follows:

$$\begin{aligned} Q(X) &= \sum_{i=1}^{r} \sum_{j=1}^{n/r} \alpha_{ij} \left(\frac{b}{r} + (-1)^i (\Delta_{i-1} + \Delta_i) + \sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik} j \delta_k + j^2 \varepsilon \right) - \beta \\ &= B + \sum_{i=1}^{r} \left((-1)^i A_i (\Delta_{i-1} + \Delta_i) + J_i \left(\sum_{k=1}^{\lfloor r/2 \rfloor} m_{ik} \delta_k \right) + \left(\sum_{j=1}^{n/r} \alpha_{ij} j^2 \right) \varepsilon \right) \\ &= B + \sum_{i=1}^{r-1} (-1)^i (A_i - A_{i+1}) \Delta_i + \sum_{k=1}^{\lfloor r/2 \rfloor} \left(\sum_{i=1}^{r} m_{ik} J_i \right) \delta_k + \sum_{i=1}^{r} \left(\sum_{j=1}^{n/r} \alpha_{ij} j^2 \right) \varepsilon. \end{aligned}$$

Finally, define

$$D_i = (-1)^i (A_i - A_{i+1}), \qquad d_k = \sum_{i=1}^{r} m_{ik} J_i, \quad \text{and} \qquad e_i = \sum_{j=1}^{n/r} \alpha_{ij} j^2,$$

for each i and k, so that

$$Q(X) = B + \sum_{i=1}^{r-1} D_i \Delta_i + \sum_{k=1}^{\lfloor r/2 \rfloor} d_k \delta_k + \sum_{i=1}^{r} e_i \varepsilon.$$

The sign of $Q(X)$ is the sign of the first nonzero coefficient in this expansion; in particular, $Q(X) = 0$ if and only if *every* coefficient B, D_i, d_k, e_i is zero. Similarly, we can write

$$Q(\hat{X}) = B + \sum_{i=1}^{r-1} D_i \Delta_i + \sum_{k=1}^{\lfloor r/2 \rfloor} d_k \delta_k + \sum_{i=1}^{r} \hat{e}_i \varepsilon,$$

where for each i,

$$\hat{e}_i = \sum_{j=1}^{n/r} \alpha_{ij} (j - p_i)^2 = e_i - 2p_i J_i + p_i^2 A_i.$$

If $B \neq 0$, the sign of B determines the sign of both $Q(X)$ and $Q(\hat{X})$. Similarly, if any of the coefficients D_i or d_k is nonzero, the first such coefficient determines the sign of both $Q(X)$ and $Q(\hat{X})$. Thus, it suffices to consider only queries for which $B = 0$, every $D_i = 0$, and every $d_k = 0$. Note that in this case, all the A_i's are equal. There are three cases to consider.

Case 1. Suppose no subset X_i contains exactly one of the query variables. (This includes the case where all query variables belong to the same subset.) Then at most $\lfloor r/2 \rfloor$ of the polynomials Q_i are not identically zero, and it follows that $A_i = 0$ for all i. The vector J consisting of the $\lfloor r/2 \rfloor$ (or fewer) nonzero J_i's must satisfy the matrix equation $(M')^\top J = 0$, where M' is a square minor of the matrix M. By condition (2) above, M' is nonsingular, so *all* the J_i's must be zero. It follows that $\hat{e}_i = e_i$ for all i, which implies that $Q(X) = Q(\hat{X})$.

Case 2. Suppose some subset X_i contains exactly one query variable t_{ij} and some other subset $X_{i'}$ contains none. Then $A_i = \alpha_{ij}$ and $A_{i'} = 0$. Since A_i and $A_{i'}$ are equal, we must have $\alpha_{ij} = 0$, but this contradicts the assumption that t_{ij} is a query variable. Thus, this case never happens.

Case 3. Finally, suppose each query variable comes from a different subset. (This includes the case of a direct query on what we claim is a critical tuple.) Recall that all the A_i's are equal. Since we are interested only in the sign of the query, we can assume without loss of generality that $A_i = \alpha_{ij} = 1$ for each query variable t_{ij}. Thus, each of the coefficients e_i is positive, which implies that $Q(X)$ is positive. Furthermore, unless the query variables are exactly x_{ip_i} for all i, each of the coefficients $\hat{e}_i$ is also positive, which means $Q(\hat{X})$ is positive.

Thus, the tuple $(x_{1p_1}, \ldots, x_{rp_r})$ is critical, as claimed. Since there are $\Omega(n^{\lceil r/2 \rceil})$ such tuples, this completes the proof of Lemma 2. □

3.2 Moving back to the reals

The presence of infinitesimals in our adversary construction means that we cannot apply our adversary argument directly, since the algorithms we consider are required to produce the correct output only when they are given real-valued input. Therefore, we must somehow eliminate the infinitesimals before applying our adversary argument. Since we know that no single real adversary input exists, we instead derive a different adversary input for each algorithm.

For any r-linear decision tree A, let Q_A denote the set of query polynomials used throughout A. We emphasize that Q_A includes *all* the polynomials used by A, not just the polynomials on any particular computation path. We can assume, without loss of generality, that Q_A includes all $\Theta(n^r)$ direct queries, since otherwise the algorithm cannot correctly detect all possible satisfying tuples.

For any input X, we call an ordered r-tuple of elements of X *relatively critical* with respect to A if the following properties are satisfied:

(1) The tuple is not in the zeroset of ϕ.

(2) There exists another collapsed input $\hat{X}$, such that the corresponding tuple in $\hat{X}$ is in the zeroset of ϕ but the sign of every other polynomial in Q_A is the same for both inputs.

Clearly, any critical tuple is also relatively critical. To prove a lower bound, it suffices to prove, for each r-linear decision tree, the existence of a corresponding nondegenerate input with several relatively critical tuples.

Lemma 3 *For any r-linear decision tree A, there exists a set $X_A \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively critical tuples.*

Proof Fix an r-linear decision tree A. Let $X \in K^n$ be given by Lemma 2. Let Φ denote the set of $\Omega(n^{\lceil r/2 \rceil})$ polynomials

$$\Phi = \left\{ \sum_{i=1}^{r} a_i t_{\pi(i)} \;\middle|\; (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(r)}) \text{ is relatively critical} \right\}.$$

Each polynomial in Φ is a direct query.

It follows directly from the definitions and Lemma 2 that the following elementary formula holds in K:

$$\exists X \bigwedge_{q \in \Phi} \left(q(X) \neq 0 \;\wedge\; \exists \hat{X} \left(q(\hat{X}) = 0 \;\wedge \bigwedge_{q' \in Q_A \setminus \{q\}} \operatorname{sgn} q'(X) = \operatorname{sgn} q'(\hat{X}) \right) \right).$$

In English, this formula reads "There is a set X such that for every direct query $q \in \Phi$, the corresponding tuple in X is relatively collapsible with respect to A."

This is just a convenient shorthand for the actual formula. The large conjunction signs are abbreviations for a conjunction of subformulas, one for each q and q'. The quantifier $\exists \hat{X}$ can be pulled out to the front of the formula, resulting in a formula with $\Omega(n^{\lceil r/2 \rceil}) + 1$ existential quantifiers, one for the original input X and one for each collapsed input $\hat{X}$.[2] The equation $\operatorname{sgn} a = \operatorname{sgn} b$ is shorthand for $((ab > 0) \vee (a = 0 \wedge b = 0))$. Finally, each reference to $q(X)$ or $q'(X)$ should be expanded into an explicit polynomial in X. (Note that within the scope of the quantifiers, X and $\hat{X}$ are formal variables.)

Since K is a subset of its real closure $\widetilde{K}$ and since the formula is only existentially quantified, the formula holds in $\widetilde{K}$. Thus, the transfer principle implies that it also holds in $\mathbb{R}$. The lemma follows immediately. □

With a little more care, we can show that the real inputs are derived by replacing the infinitesimals by sufficiently small and sufficiently well-separated real values. Thus, our infinitesimals play exactly the same role as the "inaccessibles" used by Dietzfelbinger and Maass [DM88, Die89], but we avoid having to derive explicit values based on the coefficients of the query polynomials.

[2] Actually, $\exists X$ is shorthand for a sequence of n quantifiers $\exists x_1 \, \exists x_2 \cdots \exists x_n$, and similarly for $\exists \hat{X}$.

3.3 Removing degeneracies

One final problem remains. The adversary inputs we construct (and, by implication, the real-valued inputs we obtain by invoking the previous lemma) are degenerate, at least when $r > 3$. However, in order to invoke our adversary argument we need a nondegenerate input. In simple cases, we can construct nondegenerate adversary inputs, but this becomes considerably more difficult as we consider larger values of r. Thus, instead of giving an explicit construction, we prove nonconstructively that an appropriate nondegenerate input exists.

Lemma 4 *For any r-linear decision tree A, there exists a nondegenerate set $X_A^* \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively critical tuples.*

Proof Fix an r-linear decision tree A, and as before, let Q_A denote the set of query polynomials used by A. Each polynomial in Q_A induces a hyperplane in the space $\mathbb{R}^n$; call the resulting set of hyperplanes H_A. For notational convenience, we color each hyperplane "red" if it corresponds to a direct query and "green" otherwise. Thus, an input is degenerate if and only if the corresponding point in $\mathbb{R}^n$ lies on a red hyperplane. The input X_A given by Lemma 3 corresponds to a point in some cell C_A in the arrangement of H_A, not necessarily of full dimension.

Let C be any cell in the arrangement, and let F be a facet of C. If exactly one hyperplane $h \in H_A$ contains F but does not contain C, and if h is a red hyperplane, then we say that F is a *critical facet* of C. There is a one-to-one correspondence between the critical facets of C_A and the relatively critical tuples in X_A.

Let C be any cell in the arrangement, let F be a facet of C, and let $\hat{F}$ be a critical facet of F. Since $\hat{F}$ is a ridge of C, it must be contained in exactly two facets of C. One of these facets is F; call the other one $\hat{C}$. Any hyperplane that contains two facets of a polyhedron contains the entire polyhedron [Zie94]. Thus, any hyperplane that contains $\hat{C}$ but not C also contains $\hat{F} \subset \hat{C}$ but not F. Since there is exactly one such hyperplane in H_A, it follows that $\hat{C}$ is a critical facet of C.

Since C_A has $\Omega(n^{\lceil r/2 \rceil})$ critical facets, it follows by induction that there is a full-dimensional cell C_A^* that also has $\Omega(n^{\lceil r/2 \rceil})$ critical facets.[3] We can choose X_A^* to be any point in this cell. □

[3]This conclusion is stronger than the lemma requires. It suffices that some cell that is contained only in green hyperplanes has $\Omega(n^{\lceil r/2 \rceil})$ critical facets.

This completes the proof of Theorem 1.

3.4 Rational and integer problems

The problems we consider allow the coefficients of ϕ to be arbitrary real numbers; similarly, the r-linear decision tree model allows the coefficients of the query polynomials to be arbitrary real numbers. In practice, however, all these coefficients are likely to be integers, or at least rationals. In this case, we can use the following result of Meyer auf der Heide [Mey84] to bound the number of bits required by our adversary construction.

Lemma 5 (Meyer auf der Heide) *Let H be a set of hyperplanes in $\mathbb{R}^n$, each of which has integer coefficients between $-M$ and M. Every vertex in the arrangement of H has rational coordinates $(p_1/p_0, p_2/p_0, \ldots, p_n/p_0)$, where $|p_i| \leq M^n n^{n/2}$ for all $0 \leq i \leq n$.*

Every bounded k-dimensional polyhedron has $k+1$ vertices whose centroid lies in the relative interior of the polyhedron. Thus, Lemma 5 implies that every bounded cell in the arrangement of H has an interior rational point of the form $(p_1/p_0, p_2/p_0, \ldots, p_n/p_0)$, where $|p_i| \leq M^n n^{n/2}(n+1)$ for all i. It follows that for any integer-r-linear decision tree that solves an integer-linear satisfiability problem, there is a corresponding rational adversary input, the absolute values of whose numerators and common denominator are bounded by $M^n n^{n/2}(n+1)$, where M is the absolute value of the largest coefficient of any query polynomial. (To ensure that the cell containing the adversary input is bounded, we observe that the adversary inputs we construct lie in the interior of the hypercube $[-1,1]^n$ and assume without loss of generality that every algorithm uses the $2n$ query polynomials $t_i \pm 1$.)

In the case where the polynomial ϕ and every query polynomial is homogeneous, all the query hyperplanes pass through the origin. In this case, we can produce an *integer* adversary input simply by scaling the rational input described above by its common denominator. Thus, for any homogeneous integer-r-linear decision tree that solves a homogeneous integer-linear satisfiability problem, there is a corresponding integer adversary input, the absolute value of whose elements is at most $M^n n^{n/2}(n+1)$. This compares favorably with Dietzfelbinger's adversary construction for sorting sums of $(r/2)$-tuples using direct queries, which can be realized using integers between 1 and n^n [Die89]. In fact, our construction improves Dietzfelbinger's

bound by roughly a factor of $n^{n/2-1}$, since in the cases he considers, we have $M = 1$.

4 Matching upper bounds

In this section, we show that our lower bound is as large as possible. We first describe simple algorithms to solve any r-variable linear satisfiability problem in $O(n^{(r+1)/2})$ time when r is odd and in $O(n^{r/2} \log n)$ time when r is even. To close the logarithmic gap when r is even and bigger than 2, we then derive a nonuniform algorithm whose running time is $O(n^{r/2})$.

As in the previous section, let $\phi = \sum_{i=1}^{r} a_i t_i - b$ for fixed coefficients $a_1, \ldots, a_r, b \in \mathbb{R}$. Given $x_1, \ldots, x_n \in \mathbb{R}$, our algorithms begin by constructing two multisets Y and Z, each containing $\lfloor r/2 \rfloor! \binom{n}{\lfloor r/2 \rfloor} = O(n^{\lfloor r/2 \rfloor})$ real numbers, as follows:

$$Y = \left\{ -\sum_{i=1}^{\lfloor r/2 \rfloor} a_i x_{\pi(i)} \;\middle|\; \pi : \{1, 2, \ldots, \lfloor r/2 \rfloor\} \hookrightarrow \{1, 2, \ldots, n\} \right\},$$

$$Z = \left\{ \sum_{i=1}^{\lfloor r/2 \rfloor} a_{i+\lfloor r/2 \rfloor} x_{\varpi(i)} - b \;\middle|\; \varpi : \{1, 2, \ldots, \lfloor r/2 \rfloor\} \hookrightarrow \{1, 2, \ldots, n\} \right\}.$$

(Here, $\hookrightarrow$ denotes a one-to-one function.)

First consider the case when r is odd. We begin by sorting Y and Z in time $O(n^{\lfloor r/2 \rfloor} \log n)$. Then for each input element x_i, we scan through Y and Z, looking for elements $y \in Y$ and $z \in Z$ such that $y - z = a_r x_i$. If such a pair is generated by a pair of maps π and ϖ whose images are disjoint, then $\phi(x_{\pi(1)}, \ldots, x_{\pi(\lfloor r/2 \rfloor)}, x_{\varpi(1)}, \ldots, x_{\varpi(\lfloor r/2 \rfloor)}, x_i) = 0$, so the input is degenerate. We can test this condition in constant time by storing the map used to generate each element of Y and Z and performing a simple table lookup. Performing the scan requires time $O(n^{\lfloor r/2 \rfloor})$ for each x_i, so the total running time of the algorithm is $O(n^{\lceil r/2 \rceil})$. Our algorithm can be modeled as a family of r-linear decision trees.

Now suppose r is even. The input is degenerate if and only if the multisets Y and Z share an element defined by maps π and ϖ with disjoint images. We can detect this condition by sorting $Y \cup Z$ and performing table lookups for every duplicate pair. This algorithm runs in $O(n^{r/2} \log n)$, and it can be modeled as a family of r-linear decision trees.

Our $\Omega(n^{\lceil r/2 \rceil})$ lower bound matches these upper bounds when r is odd, but it is a logarithmic factor away when r is even and greater than 2. We use the following result of Fredman [Fre76] to show that our lower bounds cannot be improved even in this case.

Lemma 6 (Fredman [Fre76]) *Let Γ be a subset of the $n!$ orderings of $\{1, 2, \ldots, n\}$ for some fixed n. There exists a comparison tree of depth at most $\log_2(|\Gamma|) + 2n$ that sorts any sequence of n numbers with order type in Γ.*

Theorem 2 *Let Π be an r-variable linear satisfiability problem with n inputs, for some fixed n and $r > 2$. There exists an r-linear decision tree with depth $O(n^{\lceil r/2 \rceil})$ that solves Π.*

Proof It suffices to show that when r is even, the multiset $Y \cup Z$ defined above can be sorted in time $O(n^{r/2})$ using Fredman's "comparison" tree, which is really an r-linear decision tree.

Every pair of elements of $Y \cup Z$ induces a hyperplane in $\mathbb{R}^n$. There is a one-to-one correspondence between the n-dimensional cells in the resulting hyperplane arrangement and the possible orderings of $Y \cup Z$. Since an arrangement of N hyperplanes in $\mathbb{R}^n$ has at most $\sum_{i=0}^{n} \binom{N}{i} = O(N^n)$ cells of dimension n (refer to [Ede87]), there are at most $O(n^{rn})$ possible orderings. It follows that the depth of Fredman's decision tree is at most $O(rn \log n) + 4(r/2)! \binom{n}{r/2} = O(n^{r/2})$. □

Of course, this result does not imply the existence of a uniform $O(n^{\lceil r/2 \rceil})$-time algorithm that works for *all* values of n. Closing the logarithmic gap between these upper and lower bounds, even for the special case of sorting $X + Y$ considered by Fredman [Fre76], is a long-standing and apparently very difficult open problem. The closest result is a simple divide-and-conquer algorithm of Steiger and Streinu [SS95], which sorts $X+Y$ in $O(n^2 \log n)$ time using only $O(n^2)$ comparisons; see also [Lam92, KK95]. Their algorithm can be adapted to solve any r-variable linear satisfiability problem, for any even $r > 2$, in $O(n^{r/2} \log n)$ time, using only $O(n^{r/2})$ r-linear queries. The extra $\log n$ factor in the running time is the result of repeating the same queries several times.

5 Conclusions and open problems

We have proven that the optimal depth of an r-linear decision tree that solves an r-variable linear satisfiability problem is $\Theta(n^{\lceil r/2 \rceil})$. Our lower bounds follow from an adversary argument. The construction of an effective adversary input is simplified by two novel techniques. First, we show that for any linear satisfiability problem, it suffices to construct a single input whose elements are taken from an extension field of the reals; Tarski's transfer principle then implies the existence of an appropriate real-valued input for each algorithm. Second, we argue that this single adversary input can be degenerate; although degenerate inputs cannot be used directly in our adversary argument, simple properties of hyperplane arrangements and convex polytopes imply the existence of appropriate nondegenerate inputs. The matching nonuniform upper bounds follow from results of Fredman [Fre76].

An obvious open problem is to improve our lower bounds to stronger models of computation. In principle, our techniques can be used to prove lower bounds in the unrestricted linear decision tree model; of course, the bottleneck is the actual adversary construction. Infinitesimal adversary constructions could also be used to prove lower bounds for higher-degree algebraic decision trees; however, the "perturbation" technique we used to prove Lemma 4 can no longer be used, since it relies crucially on properties of hyperplane arrangements and convex polytopes that are not shared by arrangements of algebraic surfaces and semialgebraic sets.[4]

Lower bounds for linear satisfiability problems in a sufficiently powerful model of computation, such as algebraic decision trees or algebraic computation trees, would imply similar lower bounds for several geometric problems, several of which we mentioned in the introduction. Although our results imply lower bounds for a few of these problems, the models in which these lower bounds hold are very weak, and most of the problems cannot even be solved in the r-linear decision tree model. Even seemingly small improvements would lead to significant new results. For example, an $\Omega(n^2)$ lower bound for 3SUM in the 6-linear decision tree model would imply the first $\Omega(n^2)$ lower bound for the problem of finding the minimum-area triangle among n points in the plane. Unfortunately, a lower bound even in the 4-linear decision tree model seems to be completely out of reach at present.

[4]Erickson and Seidel [ES95] and an earlier version of this paper both claim a generalization of our perturbation technique to restricted classes of algebraic decision trees, but the proofs are incorrect; see [ES97].

Ultimately, we would like to prove a lower bound larger than $\Omega(n \log n)$ for *any* non-NP-hard polynomial satisfiability problem, in some general model of computation such as linear decision trees, algebraic decision trees, or even algebraic computation trees. Linear satisfiability problems, in particular, 3SUM, seem to be good candidates for study.

Acknowledgments

I am deeply indebted to the anonymous referees, whose thorough reading of the article led to many significant improvements in its presentation, including the elimination of one major bug; see [ES97]. I also thank Raimund Seidel for several helpful discussions.

Acknowledgment of support

This research was done while the author was a graduate student at University of California–Berkeley, with the partial support of NSF grant CCR-9058440. Portions of this research were done at the NSF Regional Geometry Institute at Smith College, Northampton, Massachusetts, July 1993, with the support of NSF grant DMS-9013220.

References

[ACH+96] E. M. Arkin, Y.-J. Chiang, M. Held, J. S. B. Mitchell, V. Sacristan, S. S. Skeina, and T. C. Yang. On minimum-area hulls. In *Proceedings of the Fourth Annual European Symposium on Algorithms*, volume 1136 of *Lecture Notes in Computer Science*, pages 334–348. Springer-Verlag, Berlin, 1996.

[BBG94] S. Bloch, J. Buss, and J. Goldsmith. How hard are n^2-hard problems? *SIGACT News*, 25:83–85, 1994.

[BCR87] J. Bochnak, M. Coste, and M-F. Roy. *Géométrie algébraique réelle*, volume 12 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3)*. Springer-Verlag, Berlin, 1987.

[Ben83] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proceedings of the Fifteenth Annual ACM Symposium on the*

Theory of Computing, pages 80–86. ACM Press, Los Alamitos, Calif., 1983.

[BLY92] A. Björner, L. Lovász, and A. C. C. Yao. Linear decision trees: Volume estimates and topological bounds. In *Proceedings of the Twenty-fourth Annual ACM Symposium on the Theory of Computing*, pages 170–177. ACM Press, Los Alamitos, Calif., 1992.

[BvKT93] P. Bose, M. van Kreveld, and G. Toussaint. Filling polyhedral molds. In *Proceedings of the Third Workshop on Algorithms and Data Structures*, volume 709 of *Lecture Notes in Computer Science*, pages 210–221. Springer-Verlag, Berlin, 1993.

[Can88] J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on the Theory of Computing*, pages 460–467. ACM Press, Los Alamitos, Calif., 1988.

[Can93] J. Canny. Computing roadmaps of general semi-algebraic sets. *Comput. J.*, 36:504–514, 1993.

[dBdGO93] M. de Berg, M. de Groot, and M. Overmars. Perfect binary space partitions. In *Proceedings of the Fifth Canadian Conference on Computational Geometry*, pages 109–114. University of Waterloo, Waterloo, Ontario, 1993.

[Die89] M. Dietzfelbinger. Lower bounds for sorting of sums. *Theoret. Comput. Sci.*, 66:137–155, 1989.

[DL79] D. P. Dobkin and R. J. Lipton. On the complexity of computations under varying sets of primitives. *J. Comput. Syst. Sci.*, 18:86–91, 1979.

[DM88] M. Dietzfelbinger and W. Maass. Lower bound arguments with "inaccessible" numbers. *J. Comput. Syst. Sci.*, 36:313–335, 1988.

[EC91] I. Emiris and J. Canny. A general approach to removing degeneracies. In *Proceedings of the Thirty-second Annual IEEE Symposium on the Foundations of Computer Science*, pages 405–413. IEEE Computer Society, Los Alamitos, Calif., 1991.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, 1987.

[EM90] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.

[Eri95] J. Erickson. Lower bounds for linear satisfiability problems. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 388–395. 1995.

[Eri99] J. Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM J. Comput.*, 28:1198–1214, 1999.

[ES95] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. *Discrete Comput. Geom.*, 13:41–57, 1995.

[ES97] J. Erickson and R. Seidel. Erratum to "Better lower bounds on detecting affine and spherical degeneracies". *Discrete Comput. Geom.*, 18:239–240, 1997.

[Fre76] M. L. Fredman. How good is the information theory bound in sorting? *Theoret. Comput. Sci.*, 1:355–361, 1976.

[GKMS97] D. Grigoriev, M. Karpinksi, F. Meyer auf der Heide, and R. Smolensky. A lower bound for randomized algebraic decision trees. *Comput. Complexity*, 6:375, 1996/97.

[GKV97] D. Grigoriev, M. Karpinski, and N. Vorobjov. Lower bound on testing membership to a polyhedron by algebraic decision and computation trees. *Discrete Comput. Geom.*, 17:191–215, 1997.

[GO95] A. Gajentaan and M. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 5:165–185, 1995.

[GV96] D. Grigoriev and N. Vorobjov. Complexity lower bounds for computation trees with elementary transcendental function gates. *Theoret. Comput. Sci.*, 157:185–214, 1996.

[Her96] A. Hernández Barrera. Finding an $o(n^2 \log n)$ algorithm is sometimes hard. In *Proceedings of the Eighth Canadian Conference on Computational Geometry*, pages 289–294. Carleton University Press, Ottawa, Canada, 1996.

[HRR91] J. Heintz, T. Recio, and M. F. Roy. Algorithms in real algebraic geometry and applications to computational geometry. In *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, volume 6 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 137–163. American Mathematical Society, Providence, 1991.

[KK95] J. Kahn and J. H. Kim. Entropy and sorting. *J. Comput. System Sci.*, 51:390–399, 1995.

[Lam92] J.-L. Lambert. Sorting the sums $x_i + y_i$ in $O(n^2)$ comparisons. *Theoret. Comput. Sci.*, 103:137–141, 1992.

[Mat95] J. Matoušek. On geometric optimization with few violated constraints. *Discrete Comput. Geom.*, 14:365–384, 1995.

[Mey84] F. Meyer auf der Heide. A polynomial time linear search algorithm for the n-dimensional knapsack problem. *J. ACM*, 31:668–676, 1984.

[Sei97] R. Seidel. Personal communication, 1997.

[SS95] W. Steiger and I. Streinu. A pseudo-algorithmic separation of lines from pseudo-lines. *Inform. Process. Lett.*, 53:295–299, 1995.

[SY82] J. M. Steele and A. C. Yao. Lower bounds for algebraic decision trees. *J. Algorithms*, 3:1–8, 1982.

[Tar51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, 2d edition, 1951.

[Yao95] A. C.-C. Yao. Algebraic decision trees and Euler characteristics. *Theoret. Comput. Sci.*, 141:133–150, 1995.

[Yao97] A. C.-C. Yao. Decision tree complexity and Betti numbers. *J. Comput. System Sci.*, 55:36–43, 1997.

[Yap90] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40:2–18, 1990.

[Zie94] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1994.