Chapter 5

Linear Satisfiability Problems

Many computational decision problems, particularly in computational geometry, can be reduced to questions of the following form: For some fixed multivariate polynomial ϕ, given a set of n real numbers, is any subset in the zero-set of ϕ? Examples include element uniqueness ($\phi = x - y$) and 3SUM ($\phi = x + y + z$). Higher dimensional examples include the affine and spherical degeneracy problems considered in Chapters 2 and 4, and Hopcroft's point-line incidence problem, which we will consider in Chapter 6.

In this chapter, we develop general techniques for proving lower bounds on the complexity of deciding problems of this type. In particular, we examine *linear satisfiability problems*, in which the polynomial ϕ is linear. Any r-variable linear satisfiability problem can be decided in $O(n^{(r+1)/2})$ time when r is odd, or $O(n^{r/2} \log n)$ time when r is even. These are the best known upper bounds; the algorithms that achieve them were described in Section 2.1.3 (with $r = d + 1$).

We consider these problems under two models of computation, both restrictions of the linear decision tree model. In the *direct query* model, each decision is based on the sign of an assignment to ϕ by r of the input variables. In the r-*linear decision tree* model, each decision is based on the sign of an arbitrary affine combination of at most r input variables. We show that in these models, any algorithm that solves any r-variable linear satisfiability problem must perform $\Omega(n^{\lceil r/2 \rceil})$ direct queries in the worst case. This matches known upper bounds when r is odd, and is within a logarithmic factor when r is even. Moreover, results of Fredman [80] establish the existence of *nonuniform* algorithms whose running times match our lower bounds exactly.

The adversary arguments we use to establish lower bounds for these require two new tricks. The first trick is to allow our adversary configurations to contain formal infinitesimals, instead of just real numbers. Tarski's Transfer Principle implies that for any algorithm, if there is a hard configuration with infinitesimals, then a corresponding real configuration exists with the same properties. Previously, Dietzfelbinger and Maass [56, 55] used a similar technique to prove lower bounds, using "inaccessible" numbers, or numbers having "different orders of magnitude". Unlike their technique, using infinitesimals makes it possible, and indeed sufficient, to derive a *single* adversary configuration for any problem, rather than explicitly constructing a different configuration for every algorithm.

The second trick is allowing our adversary configurations to be degenerate. That is, both the original configuration and the collapsed configuration contain tuples in the zero-set of ϕ. We show that such a configuration can always be perturbed into general position, so that the new configuration has just as many collapsible tuples as the original.

An $\Omega(n \log n)$ lower bound for any linear satisfiability problem follows easily from techniques of Dobkin and Lipton in the linear decision tree model [58], Steele and Yao in the algebraic decision tree model [138], and Ben-Or in the algebraic computation tree model [16]. The first better lower bound is due to Fredman [80], who proved an $\Omega(n^2)$ lower bound on the number of comparisons required to detect duplicate elements in the Minkowski sum $X + Y$ of two sets of real numbers; his proof relies on a simple adversary argument. Fredman's result was generalized by Dietzfelbinger [55], who derived an $\Omega(n^{r/2})$ lower bound on the depth of any comparison tree algorithm that determines, given a set of n reals, whether any two 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 the direct query model, for all even r. Dietzfelbinger's results imply a lower bound in the more general r-linear decision tree model as well.

Our lower bounds should be compared with the following result of Meyer auf der Heide [114]: For any fixed n, there exists a linear decision tree of depth $O(n^4 \log n)$ that decides the n-dimensional knapsack problem. This nonuniform algorithm can be adapted to solve any of the linear satisfiability problems we consider, in the same amount of time [56]. Thus, there is no hope of proving lower bounds bigger than $\Omega(n^4 \log n)$ for any of these problems in the 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.

5.1 Preliminaries

An *ordered field* is a field with a strict linear ordering $<$ compatible with the field operations, or more abstractly, a field in which the equation $\sum_i a_1^2 = 0$ has no nontrivial solutions. 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 [22] or [124] for further details and more formal definitions, and to [25, 26] for previous algorithmic applications of real closed fields.

An *elementary formula*[1] is a finite quantified boolean formula, each of whose clauses is a multivariate polynomial inequality with real coefficients. An elementary formula *holds in* an ordered field K if and only if the formula has no free variables, and the formula is true if we interpret each variable as an element of K and addition and multiplication as field operations in K.

The following principle was originally proven by Tarski [146], in a slightly different form. See [22] for a more recent proof.

The Transfer Principle: *Let $\widetilde{K}$ and $\widetilde{K'}$ be two real closed fields. An elementary formula holds in $\widetilde{K}$ if and only if it holds in $\widetilde{K'}$.*

In particular, this implies that if an elementary formula holds in *any* real closed field, then it must hold in the reals.

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 $\mathbb{R}(\varepsilon_1, \varepsilon_2, \varepsilon_3)$, 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

[1] or more formally, a formula in the first-order language of ordered fields with parameters in $\mathbb{R}$ [22]

a_i is zero. Infinitesimals have been used extensively in perturbation techniques [67, 71, 160], in algorithms dealing with real semialgebraic sets [25, 26], and in at least one other lower bound argument [93].

Let us now formally define our model of computation. Recall that a *linear decision tree* is a ternary tree in which each interior node v in the tree is labeled with a linear *query polynomial* $q_v \in \mathbb{R}[t_1, \ldots, t_n]$ and its branches labeled -1, 0, and $+1$. Each leaf is labeled with some value; for our purposes, these values are all either "true" or "false". We compute with such a tree as follows. Given an input $X \in \mathbb{R}^n$, the sign of $q_v(X)$ is computed, where v is the root of the tree, and the computation proceeds recursively in the appropriate subtree. When a leaf is reached, its label is returned as the output of the algorithm. (Compare [58, 138].) An r-*linear* decision tree is a linear decision tree, each of whose query polynomials has at most r terms.

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 input from K^n. (We emphasize that query polynomials always have real coefficients, even when we consider more general inputs.) For any ordered field K, we will refer to the space K^n of possible inputs as the *configuration space*, and its individual elements as *configurations*.

5.2 The Lower Bound

In this section, we prove the following lower bound.

Theorem 5.1. *Any r-linear decision tree that decides an r-variable linear satisfiability problem must have depth $\Omega(n^{\lceil r/2 \rceil})$.*

Throughout this section, let ϕ denote a fixed linear expression in r variables. We say that an r-tuple is *degenerate* if it lies in the zero-set of ϕ, and that a configuration X is degenerate if it contains any degenerate r-tuples. For any configuration X, we call an r-tuple of elements of X *collapsible* if the following properties are satisfied.

(1) The tuple is nondegenerate.

(2) There exists another *collapsed* configuration $\check{X}$, such that the corresponding tuple in $\check{X}$ is degenerate, but the sign of every other real affine combination of r or fewer elements is the same for both configurations.

In other words, the only way for an r-linear decision tree to distinguish between X and $\check{X}$ is to perform a direct query on the tuple. Our usual adversary argument implies that the number of collapsible tuples in any nondegenerate configuration is a lower bound on the depth of any r-linear decision tree.

Unfortunately, this approach seems to be doomed from the start. For *any* two sets X and Y of real numbers, there are an infinite number of query polynomials that are positive at X and negative at Y. It follows that real configurations cannot contain collapsible tuples. Moreover, for any set X of n real numbers, there is an algorithm which requires only n queries to decide whether X satisfies any fixed linear satisfiability problem. Thus, no single real configuration is hard for every algorithm.

To get around this problem, we allow our adversary configurations to contain elements of an ordered field of the form $K = \mathbb{R}(\varepsilon_1, \ldots, \varepsilon_m)$. Allowing the adversary to use infinitesimals lets us construct a configuration with several collapsible tuples (Lemma 5.3), even though such configurations are impossible if we restrict ourselves to the reals.

The algorithms we consider are only required to behave correctly when they are given real input. Therefore, before applying our adversary argument, we must first eliminate the infinitesimals. The second step in our proof (Lemma 5.4) is to derive, for each r-linear decision tree, a corresponding real configuration with several *relatively* collapsible tuples (defined below). This step follows from our infinitesimal construction by a straightforward application of Tarski's Transfer Principle.

Finally, the adversary configurations we construct in the first step (and by implication, the real configurations we get by invoking the Transfer Principle) contain several r-tuples in the zeroset of ϕ. Thus, the collapsible tuples do not immediately give us the lower bound, since both the original configuration and the collapsed configuration are degenerate. In the final step of the proof (Lemma 5.5), we show that these degenerate configurations can be perturbed into general position. The lower bound then follows from our usual adversary argument.

5.2.1 The Infinitesimal Adversary Configuration

Our construction relies on an integer matrix M satisfying the following lemma.

Lemma 5.2. *There exists an $r \times \lfloor r/2 \rfloor$ integer matrix M satisfying the following two 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: Let $M = (m_{ij})$ be the $r \times \lfloor r/2 \rfloor$ integer matrix whose first $\lceil r/2 \rceil$ rows form a Vandermonde matrix with $m_{ij} = i^{j-1}$, and whose last $\lfloor r/2 \rfloor$ rows form a negative identity matrix. We claim that this matrix satisfies conditions (1) and (2).

We construct 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 $m_{max} = \lceil r/2 \rceil^{\lfloor r/2 \rfloor - 1}$ denote the largest element in M. Fix the first $\lceil r/2 \rceil$ coordinates of v arbitrarily in the range

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

Now assign the following values to the remaining $\lfloor r/2 \rfloor$ coordinates:

$$v_j = \sum_{i=1}^{\lceil r/2 \rceil} m_{i, j - \lceil r/2 \rceil} v_i.$$

Since each m_{ij} is a positive integer, the v_j are all positive integers in the range $\lceil r/2 \rceil \le v_j \le n$. We easily verify that $M^\top v = 0$. There are

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

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. Using elementary row and column operations, we can write

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

where W is a matrix with determinant ± 1, V is a square minor of a nonnegative Vandermonde matrix, and I is an identity matrix. Since W, V', and I are all nonsingular, so is M'. Thus, M satisfies condition (2). $\square$

Lemma 5.3. *There exists a configuration $X \in K^n$ with $\Omega(n^{\lceil r/2 \rceil})$ collapsible tuples, for some ordered field K.*

Proof: We explicitly construct a configuration $X \in \mathbb{R}(\Delta_1, \ldots, \Delta_{r-1}, \delta_1, \ldots, \delta_{\lfloor r/2 \rfloor}, \varepsilon_1, \ldots, \varepsilon_r)$ that satisfies the lemma. We assume without loss of generality that n is a multiple of r.

Write $\phi = \sum_{i=1}^{r} a_i t_i$ with real coefficients a_i and formal variables t_i. Let the matrix $M = (m_{ij})$ be given by the previous lemma. Our configuration X is the union of r smaller sets X_i, each containing n/r elements x_{ij} defined as follows.

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

For notational convenience, we define $\Delta_0 = \Delta_r = 0$.

We claim that any tuple $(x_{1p_1}, \ldots, x_{rp_r})$ satisfying the equation $M^\top(p_1, \ldots, p_r) = 0$ is collapsible. By condition (1), there are $\Omega((n/r)^{\lceil r/2 \rceil}) = \Omega(n^{\lceil r/2 \rceil})$ such tuples. The adversary collapses the tuple by replacing X with $\check{X}$, with elements

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

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

For example, in the simplest nontrivial case $r = 3$, our adversary configuration X lies in the field $\mathbb{R}(\Delta_1, \Delta_2, \delta_1, \varepsilon_1, \varepsilon_2, \varepsilon_3)$. If we take $M = (1, 1, -1)^\top$, then X contains the following elements, for all $1 \le j \le n/3$:

$$\begin{aligned}
x_{1j} &= (-\Delta_1 \phantom{{}+\Delta_2} + \delta_1 j + \varepsilon_1 j^2)/a_1,\\
x_{2j} &= (\Delta_1 + \Delta_2 + \delta_1 j + \varepsilon_2 j^2)/a_2,\\
x_{3j} &= (- \Delta_2 + \delta_1 j + \varepsilon_3 j^2)/a_3,
\end{aligned}$$

The indices of each allegedly collapsible tuple satisfy the equation $p_1 + p_2 = p_3$, and the corresponding collapsed configuration $\check{X}$ has the following elements:

$$\begin{aligned}
\check{x}_{1j} &= (-\Delta_1 \phantom{{}+\Delta_2} + \delta_1 j + \varepsilon_1 (j - p_1)^2)/a_1,\\
\check{x}_{2j} &= (\Delta_1 + \Delta_2 + \delta_1 j + \varepsilon_2 (j - p_2)^2)/a_2,\\
\check{x}_{3j} &= (- \Delta_2 + \delta_1 j + \varepsilon_3 (j - p_3)^2)/a_3.
\end{aligned}$$

Fix a tuple $(x_{1p_1}, \ldots, x_{rp_r})$ where $M^\top(p_1, \ldots, p_r) = 0$, and let $\check{X}$ be the corresponding collapsed configuration. We easily confirm that the collapsed tuple $(\check{x}_{1p_1}, \ldots, \check{x}_{rp_r})$ is degenerate. It remains to show that every other r-linear query expression has the same sign in both X and $\check{X}$.

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 configuration X above.

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

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

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

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

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

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

Let us 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

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

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 is zero. Similarly, we can write

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

where for each i,

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

If any of the coefficients D_i or d_k is nonzero, then the first such coefficient determines the sign of both $Q(X)$ and $Q(\check{X})$. Thus, it suffices to consider only queries for which 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 Q_i's 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 $\check{e}_i = e_i$ for all i, which implies that $Q(X) = Q(\check{X})$.

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 x_{kj} 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 collapsible tuple.) Recall that all the A_i's are equal. Since we are only interested 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 $\check{e}_i$ is also positive, which means $Q(\check{X})$ is also positive.

This completes the proof of Lemma 5.3. □

5.2.2 Moving Back to the Reals

The configurations we construct are not directly usable in an adversary argument, because the algorithms we consider are only required to be correct when given real input. Thus, before we can apply our adversary argument, we must eliminate the infinitesimals. Since we know that a single real configuration cannot be hard for *every* algorithm, we are forced to derive, for each algorithm, a corresponding real configuration that is hard for that particular algorithm. Rather than constructing such configurations explicitly in terms of the coefficients of the query polynomials, as was done in [56, 55], we nonconstructively derive their existence from our infinitesimal construction.

Let $\mathcal{Q}_{\mathcal{A}}$ denote the set of query polynomials used by any r-linear decision tree $\mathcal{A}$. (We assume, without loss of generality, that $\mathcal{Q}_{\mathcal{A}}$ includes all $\Theta(n^r)$ direct queries, since otherwise the algorithm cannot correctly detect all possible degenerate tuples.) For any input configuration X, we call an r-tuple of elements in X *relatively collapsible* if the following properties are satisfied.

(1) The tuple is nondegenerate.

(2) There exists another *collapsed* configuration $\check{X}$, such that the corresponding tuple in $\check{X}$ is degenerate, but the sign of every other polynomial in $\mathcal{Q}_{\mathcal{A}}$ is the same for both configurations.

Clearly, any collapsible tuple is also relatively collapsible. To prove a lower bound, it suffices to prove, for each r-linear decision tree $\mathcal{A}$, the existence of a corresponding nondegenerate input configuration with a large number of relatively collapsible tuples.

Lemma 5.4. *For any r-linear decision tree $\mathcal{A}$, there exists a real configuration $X_{\mathcal{A}} \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples.*

Proof: Fix $\mathcal{A}$, and let $X \in K^n$ be the configuration given by Lemma 5.3. Each of the collapsible tuples in X is clearly also relatively collapsible. Each relatively collapsible tuple Y in X corresponds to a polynomial ϕ_Y, such that $\phi_Y(X) = \phi(Y)$. Call the set of these polynomials Φ.

It follows directly from the definitions that the following elementary formula holds in K.

$$\exists X \bigwedge_{\phi_Y \in \Phi} \left(\phi_Y(X) \neq 0 \ \wedge \ \exists \check{X} \left(\phi_Y(\check{X}) = 0 \ \wedge \bigwedge_{q \in \mathcal{Q}_{\mathcal{A}} \setminus \{\phi_Y\}} \operatorname{sgn} q(X) = \operatorname{sgn} q(\check{X}) \right) \right)$$

This is just a convenient shorthand for the actual formula. Each reference to $\phi_Y(X)$ or $q(X)$ should be expanded into an explicit polynomial in X. The equation $\operatorname{sgn} a = \operatorname{sgn} b$ into the boolean formula $((ab > 0) \vee (a = 0 \wedge b = 0))$. Since the sets Φ and $\mathcal{Q}_{\mathcal{A}}$ are finite, the expanded formula is also finite and therefore elementary.

Since K is a subset of its real closure $\widetilde{K}$, and the formula is only existentially quantified, the formula also holds in $\widetilde{K}$. Thus, by the Transfer Principle, it also holds in $\mathbb{R}$. The lemma follows immediately. $\square$

With a little more care, we can show that the real configurations are derived by replacing the infinitesimals by sufficiently small and sufficiently well-separated real values, but this is not necessary to prove our lower bounds.

5.2.3 Perturbing into General Position

One final problem remains. The adversary configurations we construct (and by implication, the real configurations we get by invoking the previous lemma) are degenerate. In simple cases, we can construct nondegenerate adversary configurations, but this becomes considerably more difficult as we consider larger values of r. Instead, we show nonconstructively that the existing degenerate configurations can be perturbed into general position.

Lemma 5.5. *For any r-linear decision tree $\mathcal{A}$, there exists a nondegenerate real configuration $X^*_{\mathcal{A}} \in \mathbb{R}^n$ with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples.*

Proof: As before, let $\mathcal{Q}_{\mathcal{A}}$ denote the set of query polynomials used by $\mathcal{A}$. Each query in $\mathcal{Q}_{\mathcal{A}}$ induces a hyperplane in the configuration space $\mathbb{R}^n$, and these hyperplanes define a cell complex, called the *arrangement* [62]. Color each hyperplane "red" if it corresponds to a direct query, and "green" otherwise.

Each input configuration corresponds to a point in some cell $\mathcal{C}$ in this arrangement. Nondegenerate configurations correspond to points in n-dimensional cells; degenerate configurations correspond to points in cells of lower dimension. As long as we never change the result of any query in $\mathcal{Q}_{\mathcal{A}}$, changing the entries in a configuration corresponds to moving the configuration point within $\mathcal{C}$. Collapsing a collapsible tuple moves the configuration point onto a boundary facet of $\mathcal{C}$ uniquely spanned by a red hyperplane. (That is, the red hyperplane is the only hyperplane that contains the facet but not the entire cell.) To prove the lemma, it suffices to find a full-dimensional cell with $\Omega(n^{\lceil r/2 \rceil})$ red boundary facets.

Let $\mathcal{C}$ be an arbitrary cell, and let $\mathcal{C}'$ be one of the cells in its boundary. Any hyperplane that uniquely spans a facet of $\mathcal{C}'$ also uniquely spans a facet of $\mathcal{C}$. Thus, if there is a cell of *any* dimension with $\Omega(n^{\lceil r/2 \rceil})$ red boundary facets, then there must be an n-dimensional cell with $\Omega(n^{\lceil r/2 \rceil})$ red boundary facets. Since relatively collapsible tuples correspond to red boundary facets, it suffices to show that there exists a (possibly degenerate) real configuration with $\Omega(n^{\lceil r/2 \rceil})$ relatively collapsible tuples. Such a configuration is guaranteed by Lemma 5.4. □

This lemma, together with our usual adversary argument, completes the proof of Theorem 5.1.

5.3 Matching Nonuniform Upper Bounds

Our lower bound matches known upper bounds when r is odd, but is a logarithmic factor away when r is even and greater than 2. We use the following result of Fredman [80] to show that our lower bounds cannot be improved in this case.

Lemma 5.6 (Fredman). *Let Γ be a subset of the $n!$ orderings of $\{1, \ldots, n\}$ 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 5.7. *For any n and $r > 2$, there exists an r-linear decision tree with depth $O(n^{\lceil r/2 \rceil})$ that solves any r-linear satisfiability problem with n inputs.*

Proof: It suffices to consider the case when r is even, since for any odd r there is a simple uniform algorithm with running time $O(n^{(r+1)/2})$. Suppose we are trying to satisfy the equation $\sum_{i=1}^{r} a_i t_i = 0$ for some fixed coefficients $a_i \in \mathbb{R}$. Given a configuration $X = (x_1, \ldots, x_n) \in \mathbb{R}^n$, we (implicitly) construct sets J and K of $n^{\underline{r/2}}$ real numbers each[2], as follows:

$$J = \left\{ \sum_{i=1}^{r/2} a_i x_{j_i} \;\middle|\; \{j_1, j_2, \ldots, j_{r/2}\} \subset \{1, 2, \ldots, n\} \right\}$$
$$K = \left\{ \sum_{i=1}^{r/2} -a_{i+r/2} x_{k_i} \;\middle|\; \{k_1, k_2, \ldots, k_{r/2}\} \subset \{1, 2, \ldots, n\} \right\}$$

Then X is degenerate if and only if the sets J and K share an element defined by tuples whose index sets $\{j_i\}$ and $\{k_i\}$ are disjoint. We can detect this condition by sorting $J \cup K$ using Fredman's "comparison" tree, which is really a r-linear decision tree.

Every pair of elements of $J \cup K$ induces a hyperplane in the configuration space $\mathbb{R}^n$. There is a one-to-one correspondence between the cells in the resulting hyperplane arrangement and the possible orderings of $J \cup K$. Since an arrangement of N hyperplanes

[2] For any integer $a \geq 0$, the falling factorial power $n^{\underline{a}}$ is defined as $n(n-1)\cdots(n-a+1) = n!/(n-a)!$ [92].

in $\mathbb{R}^D$ has at most $\sum_{i=0}^{D}\binom{N}{i} = O(N^D)$ cells [62], there are at most $O((2n^{\underline{r/2}})^{2n}) = O((2n)^{rn})$ possible orderings. It follows that the depth of Fredman's decision tree is at most $4n^{\underline{r/2}} + O(rn\log n) = O(n^{r/2})$. □

Of course, this result does not imply the existence of a single $O(n^{\lceil r/2\rceil})$-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 the Minkowski sum $X+Y$ of two sets, is a long-standing and very difficult open problem. The closest result is an algorithm of Steiger and Streinu [139] that sorts $X+Y$ in $O(n^2\log n)$ time using only $O(n^2)$ comparisons.

5.4 Conclusions and Open Problems

We have developed a new general technique for proving lower bounds in decision tree models of computation. We show that it suffices to construct a single input configuration, possibly degenerate and possibly containing infinitesimals, containing several collapsible tuples. Using this technique, we have proven $\Omega(n^{\lceil r/2\rceil})$ lower bounds on the depth of any r-linear decision tree that decides an r-variable linear satisfiability problem. This is the best possible lower bound in this model.

An immediate open problem is to improve our lower bounds to stronger models of computation. It seems "obvious" that linear queries with more variables or higher-degree queries almost never give us useful information, and therefore can almost always be added to our model of computation with impunity. Can we define a general class of "allowable" queries for linear satisfiability problems, as we did in the previous chapters?

Can the techniques developed in this chapter be applied to higher-order polynomial satisfiability problems?

There are simple reductions from linear satisfiability problems to many other higher-dimensional geometric problems. For example, finding a d-tuple in the zeroset of the polynomial $\sum_{i=1}^{d} t_i$ can be reduced to the d-dimensional affine degeneracy problem. (See Lemma 2.3.) Several more good examples can be found in Gajentaan and Overmars' collection of 3SUM-hard problems [86]. Unfortunately, these reductions use primitives disallowed by the models of computation in which our lower bounds hold. Consequently, our linear satisfiability lower bounds do *not* imply lower bounds for these geometric problems.

Can the techniques of this chapter be applied directly to higher-dimensional problems? (My original presentation of these results [74] claimed to do just that, but the proofs were flawed; Lemma 4.1 in [74] is actually false.)

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 seem to be good candidates for study.

"Now, even though their jumping is blind and wholly random, there are billions upon billions of atoms in every interstice, and as a consequence of this great number, their little skips and scamperings give rise to, among other things—and purely by accident—to significant configurations.... Do you know what a configuration is, blockhead?"

"No insults, please!" said Pugg. "For I am not your usual uncouth pirate, but refined and with a Ph.D., and therefore extremely high-strung."

— Stanislaw Lem (translated by Michael Kandel), "The Sixth Sally, or How Trurl and Klapaucius Created a Demon of the Second Kind to Defeat the Pirate Pugg", *The Cyberiad*, 1974