Applications of Ramsey's Theorem to Decision Tree Complexity

SHLOMO MORAN

Technion, Haifa, Israel

MARC SNIR

Hebrew University, Jerusalem, Israel

AND

UDI MANBER

University of Wisconsin—Madison, Madison, Wisconsin

Abstract. Combinatorial techniques for extending lower bound results for decision trees to general types of queries are presented. Problems that are defined by simple inequalities between inputs, called *order invariant* problems, are considered. A decision tree is called *k-bounded* if each query depends on at most k variables. No further assumptions on the type of queries are made. It is proved that one can replace the queries of any k-bounded decision tree that solves an order-invariant problem over a large enough input domain with k-bounded queries whose outcome depends only on the relative order of the inputs. As a consequence, all existing lower bounds for comparison-based algorithms are valid for general k-bounded decision trees, where k is a constant.

An $\Omega(n \log n)$ lower bound for the element uniqueness problem and several other problems for any k-bounded decision tree, such that $k = O(n^c)$ and $c < \frac{1}{2}$ is proved. This lower bound is tight since there exist $n^{1/2}$-bounded decision trees of complexity $O(n)$ that solve the element-uniqueness problem. All the lower bounds mentioned above are shown to hold for nondeterministic and probabilistic decision trees as well.

Categories and Subject Descriptors: F.1.2 **[Computation by Abstract Devices]**: Modes of Computation—*probabilistic computation*; F.2.2 **[Analysis of Algorithms and Problem Complexity]**: Nonnumerical Algorithms and Problems—*computations on discrete structures, sorting and searching*; G.2.1 **[Discrete Mathematics]**: Combinatorics—*combinatorial algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Computational complexity, decision trees, lower bounds, Ramsey's theorem

1. *Introduction*

Decision trees are very useful in proving lower bounds for combinatorial problems [2, 5, 6, 10, 14, 15, 22]. In particular, they have been extensively used to analyze sorting-type problems whose outcome depends on the relative order of the inputs.

An earlier version of this paper was presented at the 25th Annual Symposium on the Foundations of Computer Science. The work of the third author was supported in part by the National Science Foundation under Grant MCS83-03134.

Authors addresses: S. Moran, Department of Computer Science, Technion, Haifa, Israel; M. Snir, Department of Computer Science, Hebrew University, Jerusalem, Israel; U. Manber, Department of Computer Science, University of Wisconsin—Madison, 1210 West Dayton Street, Madison, WI 53706.

One weakness of many results is the restriction on the type of queries that can be performed. It is only the *information-theoretic* lower bound that is valid with no restrictions on the type of queries used. However, the information-theoretic argument does not yield useful lower bounds for many problems, in particular, recognition problems that have only two outcomes. Examples of lower bounds that are not information theoretic are the $n-1$ lower bound for maximum finding, the lower bounds for selection and merging, and the $\Omega(n \log n)$ lower bound for element uniqueness.

Significant amount of work has been done in extending these lower bounds to decision trees with less restricted queries. Thus, Reingold [13] extended the $n-1$ lower bound for maximum finding to decision trees using linear comparisons; Yao [20] and Dobkin and Lipton [2] did the same for the selection problem and the element-uniqueness problem, respectively. Rabin [11] extended the lower bound for maximum finding to decision trees using comparisons of meromorphic functions. Ben-Or [1] extended lower bounds for several problems to bounded-degree algebraic decision trees (see also [19]). Manber and Tompa [9] extended several lower bounds to nondeterministic and probabilistic models of decision trees (see also [8] and [18]).

All these results assume that the inputs are taken from R, the set of real numbers. This allows the authors to use sophisticated geometric tools. On the other hand, the purely combinatorial nature of the original problems is lost.

In this paper we present combinatorial techniques for extending lower bounds for decision trees to general types of queries. At the heart of the techniques is the use of Ramsey's theorem. We consider problems, which we call *order invariant*, that are defined by simple inequalities between inputs. These are precisely the problems that can be solved by decision trees using comparisons of the form $x_i : x_j$. A query is *order invariant* if its outcome depends only on the relative order of the inputs occurring in it. The *arity* of a query is the number of inputs that the query depends on. We assume that inputs are drawn from a large finite (or infinite) totally ordered set. We make no further assumptions on the set of inputs or the type of queries.

We prove the following result: Let T be a decision tree that solves an order-invariant problem over a large enough input domain. Then each query in T can be replaced by an order-invariant query of the same arity, such that the resulting decision tree still solves the original problem.

A decision tree is called *k-bounded* if each query depends on at most k variables. The last result implies that decision trees that use only simple comparisons between inputs are as powerful as 2-bounded decision trees for solving order-invariant decision problems. Up to a constant factor, the same claim holds for k-bounded decision tree, where k is a constant. Thus, all existing lower bounds for comparison-based algorithms are valid for general k-bounded decision trees.

Decision trees using linear comparisons are known to be more powerful than decision trees using simple comparisons in solving certain order-invariant problems (Snir, [16]). The last result shows that the discrepancy is due uniquely to the fact that a linear comparison may involve many inputs, whereas a simple comparison involves only two inputs.

We also prove lower bounds for specific problems allowing general queries with nonconstant arity. We use the combinatorial techniques developed in [9] for probabilistic decision trees and extend them by using Ramsey's theorem. We prove an $\Omega(n \log n)$ lower bound for the element uniqueness problem for any k-bounded decision tree, such that $k = O(n^c)$ and $c < \frac{1}{2}$. This is a tight result in the sense that,

if $k = n^{1/2}$, then there exist k-bounded decision trees of complexity $O(n)$ that solve the element uniqueness problem. In proving this, we use Ramsey's theorem in a more direct way. This makes the results valid for input domains that are much smaller than the input domains required for the more general results (although they are still quite large). The $\Omega(n \log n)$ lower bound applies to other problems such as set equality, set disjointedness, and ϵ-closeness [5].

Both results are extended to nondeterministic decision trees and probabilistic decision trees using the techniques of [9] and [18].

Recently, Maass [7] independently found similar techniques using Ramsey's theorem to prove lower bounds for random-access machines.

2. *Definitions*

Let S be a totally ordered set and n a positive integer. Let S^n denote the set of all n-tuples of elements of S, and let $[S]^n$ denote the set of all n-subsets of S. A *decision problem* Δ is a partition $D_1, \ldots, D_q$, of S^n (the problem is to determine to which set D_i an input belongs). Two tuples $\mathbf{x}$ and $\mathbf{y}$ are *order equivalent*, $\mathbf{x} \equiv \mathbf{y}$, if for each i and j, $x_i < x_j \Leftrightarrow y_i < y_j$. We call the equivalence class of $\mathbf{x}$ the *order type* of $\mathbf{x}$. If π is a permutation of n elements, then the *order type of* π *in* S, denoted by S_π, is the set of all tuples $(a_1, \ldots, a_n) \in S^n$ in which $a_i < a_j$ iff $\pi(i) < \pi(j)$. A decision problem Δ is *order invariant* if each set D_i of the partition is closed under the equivalence relation $\equiv$; Δ is order invariant iff each set D_i can be defined by Boolean combinations of assertions of the form $x_i < x_j$.

A *deterministic decision tree* T is a labeled binary tree. Each internal node v of T is labeled with a query Q_v, which is a predicate defined on S^n. The two outgoing edges of v are labeled by T(rue) or F(alse). Each leaf is labeled by one of the sets of the partition Δ. The predicates are defined on the whole set S^n for simplicity of notation. We associate with each predicate Q a set of indices $I_Q = \{i_1, i_2, \ldots, i_r\}$, such that, given an input $(x_1, x_2, \ldots, x_n)$, the value of Q depends only on $(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$. The parameter r above is the *arity of* Q.

The evaluation of T on an input $\mathbf{x}$ proceeds downward from the root. If the node v is reached, then the predicate Q_v is evaluated on $\mathbf{x}$, and one of the outgoing edges is chosen according to the outcome of the evaluation. The path $\mathbf{x}$ follows is called the *computation path* for $\mathbf{x}$. The tree T *solves* Δ *on* C if, for each $\mathbf{x} \in C$, $\mathbf{x}$ reaches a leaf with label D_i iff $\mathbf{x} \in D_i$; T solves Δ if it solves it on S^n, the domain of the problem.

We consider, in particular, *recognition problems*, that is, decision problems that have two outcomes only. In that case we label the leaves with *accept* and *reject*; the set accepted by T consists of the elements of S^n whose computation path terminates in an accepting leaf, and such a path is an *accepting path.*

A *probabilistic decision tree* with one-sided error [9] is a decision tree that also has some internal nodes that are *coin-tossing* nodes. When a computation path reaches such a node, it takes either of the emanating edges with probability $\frac{1}{2}$. The set accepted by such a tree is the set of inputs with a positive probability of being accepted. We require that, if an input is accepted, then it is accepted with probability $\geq \frac{1}{2}$. A probabilistic decision tree with two-sided error is a probabilistic decision tree with a slightly different accepting rules. The accepted set is the set of inputs with probability $\geq \frac{3}{4}$ of reaching an accepting leaf, and we also require that all the other inputs have probability $\geq \frac{3}{4}$ of reaching a rejecting leaf.

A predicate Q is *order invariant* if its truth set is order invariant, that is, $\mathbf{x} \equiv \mathbf{y} \Rightarrow Q(\mathbf{x})$ iff $Q(\mathbf{y})$; Q *is order invariant on* C if $\mathbf{x}, \mathbf{y} \in C$, $\mathbf{x} \equiv \mathbf{y} \Rightarrow Q(\mathbf{x})$ iff $Q(\mathbf{y})$. The

decision tree T *is order invariant on* C if each predicate occurring in T is order invariant on C.

T is called *k-bounded* if the maximal arity of a predicate occurring in T is k. The *height* of a tree T, denoted by $h(T)$, is equal to the length of the longest path in T; the *k-complexity* $C_k(\Delta)$ of Δ is equal to the least height of a k-bounded binary decision tree that solves Δ; the *k-restricted complexity* $Cr_k(\Delta)$ of Δ is equal to the least height of a k-bounded binary decision tree that is order invariant and solves Δ. It was shown in [9] that it is sufficient to consider the height as a measure for time complexity for probabilistic decision trees as well.

3. *Lower Bounds for Constant-Bounded Decision Trees*

We prove in this section that order-invariant decision trees are as powerful as general-decision trees in solving order-invariant problems. The proof consists of two parts. The first (easy) part, consists of showing that, if an order-invariant decision tree solves an order-invariant decision problem on a set of inputs that contains representatives of each order type, then it solves the problem correctly for any input. In the second (harder) part, we show that, if T is a decision tree that solves an order-invariant problem Δ defined on S^n, and S is large enough, then there exist a subset $C \subset S$ such that C contains at least n elements, and T is order invariant for inputs from C^n. Ramsey's theorem is used to prove that claim. It follows that the predicates labeling the nodes of T can be replaced by order-invariant predicates so that the resulting decision tree still solves the initial decision problem on C. As each order type is represented in C^n, the new decision tree solves the problem Δ correctly for any input.

LEMMA 3.1. $Cr_k(\Delta) \leq Cr_2(\Delta) \leq O(k \log k) Cr_k(\Delta)$.

PROOF. The left inequality is immediate. To prove the right inequality, note that the order type of a k-tuple can be determined in $O(k \log k)$ comparisons (e.g., by sorting the tuple, next checking for equalities between successive items). But the value of an order-invariant predicate is uniquely determined by the order type of its argument. Thus, if T is a k-bounded, order-invariant decision tree, then we can replace each node v of T by a 2-bounded, order-invariant tree of height $O(k \log k)$, suitably replicating the left and right subtrees at v, so that the resulting tree T' yields the same answers as T. The decision tree T' is a 2-bounded, order-invariant tree, and $h(T') = O(k \log k) h(T)$. □

LEMMA 3.2. *Let* $\Delta = \{D_1, \ldots, D_q\}$ *be an order-invariant problem defined on* S^n, *and let* $F \subset S^n$ *be a set that contains a representative for each order type. Let* T *be an order-invariant decision tree that solves* Δ *on* F. *Then* T *solves* Δ.

PROOF. Let $\mathbf{x} \in S^n$, and assume $\mathbf{x} \in D_i$. Let $\mathbf{y} \in F$ be order equivalent to $\mathbf{x}$. Then $\mathbf{x} \in D_i$, and $\mathbf{y}$ reaches in T a leaf labeled with D_i. But $\mathbf{y}$ reaches the same leaf of T as $\mathbf{x}$. Hence, $\mathbf{x}$ reaches in T a leaf with label D_i. □

We make use of the following well-known theorem [12].

RAMSEY'S THEOREM. *For any* n, m, *and* q, *there exists a number* $N(n, m, q)$ *such that the following is true: Let* S *be a set of size at least* $N(n, m, q)$; *if we divide* $[S]^n$ *into* q *parts, then at least one part contains all of* $[C]^n$ *for some set* $C \subset S$ *of size* m.

The following theorem is also due to Ramsey [12].

THEOREM 3.3. *For any k, m, and n, there exists a number $N(k, m, n)$ such that the following is true: Let S be a totally ordered set of size at least $N(k, m, n)$; let $P_1, \ldots, P_k$ be k predicates defined on S^n. Then there exists a subset $C \subset S$ of size at least m such that each predicate P_i is order invariant on C^n.*

PROOF. Let $\{x_1, \ldots, x_r\}$ and $\{y_1, \ldots, y_r\}$ be two r-element subsets of S, indexed in increasing order. We say that $\{x_1, \ldots, x_r\}$ is *congruent* to $\{y_1, \ldots, y_r\}$ if, for each mapping σ: $1 \ldots n \to 1 \ldots r$, and each $1 \le j \le k$

$$P_j(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) \quad \text{iff} \quad P_j(y_{\sigma(1)}, \ldots, y_{\sigma(n)}).$$

It is easy to see that this is indeed an equivalence relation on $[S]^r$. The number G of equivalence classes of this relation is bounded by 2^{kr^n}. According to Ramsey's theorem, for any s there is a number $N = N(k, s, G)$ such that, if $|S| \ge N$, then S contains a subset S' such that $|S'| \ge s$ and all elements of $[S']^k$ belong to the same congruence class.

If S is large enough, we can repeat this process for $k = 1, \ldots, n$, thus building a sequence of sets $S = C_0 \supset C_1 \supset \cdots \supset C_n = C$, such that $|C| \ge m$, and all elements of $[C_k]^k$ are congruent, $k = 1, \ldots, n$.

Let $\mathbf{x}$ and $\mathbf{y}$ be two order-equivalent tuples in C^n. Let $x'_1, \ldots, x'_k$ be the distinct components of $\mathbf{x}$, indexed in increasing order, and let $y'_1, \ldots, y'_k$ be similarly defined for $\mathbf{y}$. Let σ: $1 \ldots n \to 1 \ldots k$ be such that $x_i = x'_{\sigma(i)}$, $i = 1, \ldots, n$. Since $\mathbf{x} \equiv \mathbf{y}$, it follows that $y_i = y'_{\sigma(i)}$, $i = 1, \ldots, n$. Since $\{x'_1, \ldots, x'_k\}$ is congruent to $\{y'_1, \ldots, y'_k\}$, $P_j(\mathbf{x})$ iff $P_j(\mathbf{y})$, for $j = 1, \ldots, k$. □

COROLLARY 3.4. *For each m, n, and t, there exists a number $M = M(m, n, t)$ such that the following holds: Let T be a binary decision tree of height t defined on inputs from S^n, where $|S| \ge M$. There exists a set $C \subset S$ such that $|C| \ge m$, and T is order invariant on C^n.*

PROOF. Follows immediately from the previous theorem. □

THEOREM 3.5. *For each m, n, k, and t, there exists a number $M = M(m, n, k, t)$ such that the following holds. Let Δ be an order-invariant decision problem defined on S^n and let T be k-bounded decision tree of height t that solves Δ. Let $|S| \ge M$. Then the predicates labeling the nodes of T can be modified so that the resulting decision tree T' is order invariant and solves Δ.*

PROOF. According to Corollary 3.4, if S is large enough, then there exists a set C such that $|C| \ge n$ and T is order invariant when restricted to inputs from C^n. Each tuple $\mathbf{x} \in S^n$ is order equivalent to a tuple $\mathbf{y} \in C^n$ (since $|C| \ge n$). Replace each predicate Q occurring in T by the predicate Q' defined as follows: $Q'(\mathbf{x}) = Q(\mathbf{y})$, where $\mathbf{y} \in C^n$ is order equivalent to $\mathbf{x}$. By the previous remark, such tuple $\mathbf{y}$ exists. As $\mathbf{Q}$ is order invariant on C^n, the definition does not depend on the choice of $\mathbf{y}$, and Q' is order invariant. Also, if Q depends only on k variables, then so does Q'.

Let T' be the decision tree obtained from T by that substitution. Then T' is k-bounded and order invariant on C. Also, if $\mathbf{x} \in C^n$, then $\mathbf{x}$ reaches a leaf v in T' iff it reaches it in T. Thus T' solves D on C^n, and by Lemma 3.2, solves (all) Δ. □

COROLLARY 3.6. *Let Δ be an order-invariant problem. Then*

(*i*) $Cr_k(\Delta) = C_k(\Delta)$;
(*ii*) $Cr_k(\Delta) \le C_2(\Delta) \le O(k \log k) Cr_k(\Delta)$.

PROOF. The first claim follows immediately from the last theorem. The second claim follows from the first claim and from Lemma 3.1. □

COROLLARY 3.7. *The results in Corollary* 3.6 *hold for probabilistic and nondeterministic decision trees as defined in* [9].

We discuss probabilistic decision trees in more detail in the next section.

4. *Lower Bounds for General k-Bounded Decision Trees*

In this section we prove lower bounds for specific problems allowing general queries with nonconstant arity. We mainly consider the problem of element uniqueness (EU), which is to decide, given n elements of S, whether they are pairwise distinct. The same technique applies to several other problems. We start with the deterministic model and then extend the results to the probabilistic model.

4.1 DETERMINISTIC DECISION TREES. Let p be a computation path in a deterministic decision tree T, and let π be a permutation. S_p is the set of all input tuples whose computation path is p. S_π is the set of all inputs of order type π. $S_{p,\pi}$ is the set of all inputs in $S_p \cap S_\pi$ and $P(\pi)$ is the minimal set of paths such that $\bigcup_{p \in P(\pi)} S_{p,\pi} = S_\pi$.

A computation path p is *complete* for π if, for every pair (i, j) such that $\pi(i) + 1 = \pi(j)$, there is a node v in p such that $i, j \in I_{Q_v}$ (i.e., Q_v depends both on x_i and x_j). Intuitively, this means that every pair of consecutive elements in π is compared in p. *p is incomplete* for π *at* i if no node on p satisfies the above for i.

LEMMA 4.1. *Let p be a computation path such that, for some i and j ($1 \le i < j \le n$), there is no query in p that involves both x_i and x_j. Assume further that the sequences*

$$s_1 = (a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_{j-1}, c, a_{j+1}, \ldots, a_n)$$

and

$$s_2 = (a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_{j-1}, e, a_{j+1}, \ldots, a_n)$$

are in S_p. Then the sequence

$$s_3 = (a_1, \ldots, a_{i-1}, d, a_{i+1}, \ldots, a_{j-1}, c, a_{j+1}, \ldots, a_n)$$

is also in S_p.

PROOF. Let s be an input tuple. Then $s \in S_p$ iff $Q(s) = Q(s_1)$ for every Q in p. In particular, $Q(s_1) = Q(s_2)$ for every Q in p. We prove that $Q(s_3) = Q(s_1)$ for every Q in p.

Let Q be a query in p. There are two cases to consider:

(1) i is not in I_Q. Then $Q(s_1) = Q(s_3)$ since Q involves only variables that get in s_3 the same value they get in s_1.
(2) i is in I_Q. Then j is not in I_Q; hence, $Q(s_2) = Q(s_3)$ by the same argument as in (1), and the claim holds since $Q(s_1) = Q(s_2)$. □

THEOREM 4.2. *Assume that $|S| \ge N(n, n+1, n!)$. If the decision tree T accepts EU and T has at most $n!$ accepting paths, then, for every permutation π, there is a path p in T that is complete for π.*

PROOF. Let π be a given permutation. There is a natural 1–1 mapping μ from S_π *onto* $[S]^n$, which maps each sequence of n (distinct) elements in S_π on the set containing these elements in $[S]^n$. Using this mapping, we associate with each decision tree T and with each permutation π a χ-coloring of $[S]^n$, where χ is the cardinality of $P(\pi)$, in the following way: Let $p_1, \ldots, p_\chi$ be the paths in $P(\pi)$. (Note

that if the set accepted by the decision tree is EU, then all these paths are accepting paths.) Color each set $\{a_1, \ldots, a_n\}$ by the integer i such that $\mu^{-1}(\{a_1, \ldots, a_n\})$ is in S_{p_i}. Since, by the definition of $P(\pi)$,

$$\bigcup_{p \in P(\pi)} S_{p,\pi} = S_\pi,$$

this coloring is a χ-coloring of $[S]^n$. Thus, by Ramsey's theorem, there is a subset S_0 of S such that $|S_0| \geq n + 1$ and all the sets of $[S_0]^n$ are colored by the same color, which means that all the sequences in $S_{0\pi}$ are in the set S_{p_k} for some path p_k. Let $p_k = p$; then we claim that p is complete for π. For simplicity, assume that $\pi = (1, 2, \ldots, n)$ (the identity permutation), and, for contradiction, assume that p is incomplete for π at some $i < n$. Let $\{d_1, d_2, \ldots, d_{n+1}\}$ be $n + 1$ distinct elements in S_0, $d_i < d_{i+1}$ for $1 \leq i \leq n$. Then since both

$$s_1 = (d_1, \ldots, d_{i-1}, d_i, d_{i+1}, d_{i+3}, \ldots, d_{n+1})$$

and

$$s_2 = (d_1, \ldots, d_{i-1}, d_{i+1}, d_{i+2}, d_{i+3}, \ldots, d_{n+1})$$

are in $S_{0\pi}$, both are also in S_p. Thus, we can apply Lemma 4.1 with $j = i + 1$, $(b, c) = (d_i, d_{i+1})$, and $(d, e) = (d_{i+1}, d_{i+2})$ to conclude that

$$s_3 = (d_1, \ldots, d_{i-1}, d_{i+1}, d_{i+1}, d_{i+3}, \ldots, d_{n+1})$$

is also in S_p. But this contradicts the assumption that T accepts EU, since p is an accepting path and s_3 should be rejected. □

LEMMA 4.3. *For each ϵ there is an n_ϵ such that, if $n > n_\epsilon$, then $\log(n!) > (1 - \epsilon) n \log n$.*

PROOF. Straightforward from Stirling's formula. □

The following definitions and Lemmas are similar to those in [9] and are outlined here. Given a computation path p, $G(p) = (V, E)$ is an undirected graph such that $V = \{1, \ldots, n\}$ and $E = \{(i, j) \mid$ there is a query in p that involves both x_i and $x_j\}$. A Hamiltonian path in $G(p)$ is a sequence of edges $[(\pi(1), \pi(2)), \ldots, (\pi(n - 1), \pi(n))]$, where π is any permutation of $\{1, \ldots, n\}$. (By defining the path as a sequence, we distinguish between the start node and the end node.) If a path p is complete for t permutations, then $G(p)$ contains t Hamiltonian paths.

LEMMA 4.4. *The number of Hamiltonian paths on a graph on n vertices $(n > 1)$ and m edges is at most $n(m/(n - 1))^{n-1}$.*

PROOF. Denote by $h(n, m)$ the maximal possible number of Hamiltonian paths starting at a given vertex in a graph on n vertices and m edges. We prove by induction on n that $h(n, m) \leq (m/(n - 1))^{n-1}$, which clearly implies the Lemma. For $n = 2$, $h(n, m) = m$, and equality holds. Assume that the Lemma holds for all graphs with n nodes, and let $G = (V, E)$ be a graph with $|V| = n + 1$, $|E| = m$. Let $v \in V$ be of degree d $(d > 0)$. Then $G(V - \{v\})$, the graph induced by G on $V - \{v\}$, has n vertices and $m - d$ edges. The number of Hamiltonian paths beginning with a given edge emanating from v is at most $h(n, m - d)$. Thus, the total number of Hamiltonian paths starting at v is at most $d \cdot h(n, m - d)$, which by the induction hypothesis is at most

$$d\left(\frac{m - d}{n - 1}\right)^{n-1}. \tag{4.1}$$

For a fixed n and m, (4.1) attains its maximum when $d = (m - d)/(n - 1)$, that is, when $d = m/n$ (this is easily verified by differentiating with respect to d). Thus, the number of Hamiltonian paths starting at v is at most $(m/n)^n$. The Lemma follows. □

THEOREM 4.5. *There exists a function $N = N(\epsilon, n)$ such that any $n^{1/2-\epsilon}$-bounded decision tree T that recognizes EU on a set S such that $|S| \geq N$ has height $\Omega(n \log n)$.*

PROOF. Let T be a decision tree that solves EU. If T is k-bounded and of height h, then, for each p, $G(p)$ contains at most $h(\frac{k}{2})$ edges. Let χ denote the number of computation paths in T and let t be a bound on the number of permutations for which a single path p in T is complete. By Theorem 4.2, $\chi \cdot t \geq n!$. Taking into account that $\chi \leq 2^h$, and using Lemma 4.4 to bound t, we obtain

$$2^h n \left[\frac{(\frac{k}{2})h}{n-1}\right]^{n-1} \geq n!.$$

Taking logarithms, we get that for all ϵ and for large enough n's

$$h + \log n + (n-1)\left[2\log k + \log\left(\frac{h}{n-1}\right)\right] \geq \log(n!) > (1-\epsilon)n\log n.$$

Assume now that $\log k = (\frac{1}{2} - \epsilon)\log n$. By rearranging terms we get

$$h + (n-1)\log\left(\frac{h}{n-1}\right) > \epsilon(n-2)\log n,$$

which can be shown to imply (for large enough n) that $h > \frac{1}{2}\epsilon n \log n$. The Theorem follows. □

THEOREM 4.6. *There exist $n^{1/2}$-bounded deterministic decision trees of height $O(n)$ that solve the element-uniqueness problem.*

PROOF. Divide the n elements into $2\lceil n^{1/2}\rceil$ blocks of size $\leq \frac{1}{2}\lceil n^{1/2}\rceil$, and check for every pair of blocks whether their union is pairwise distinct. There are $O(n)$ pairs of blocks, and it is easy to see that each pair of elements is contained in one such union. □

COROLLARY 4.7. *The complexity of k-bounded deterministic decision trees, where $k \leq n^{1/2-\epsilon}$, for the following problems is $\Omega(n \log n)$: set equality, set disjointness, ϵ-closeness.*

PROOF. The proofs are very similar to the proof of Theorem 4.5 and will be omitted here. The reader is referred to [9] for more details. □

4.2 PROBABILISTIC DECISION TREES. We now consider probabilistic decision trees and show that the results obtained in Theorem 4.5 hold for this model as well. The next theorem is an extension of [9, Theorem 8], where the same lower bound was proved for probabilistic decision trees with only simple comparisons.

THEOREM 4.8. *There exists a function $N = N(\epsilon, n)$ such that any $n^{1/2-\epsilon}$-bounded two-sided probabilistic decision tree T that recognizes EU on a set S, such that $|S| \geq N$, has height $\Omega(n \log n)$.*

PROOF. Let T be two-sided error probabilistic decision tree that solves EU. A path p in T is called *half-complete* for a permutation π if the number of pairs (i, j), $\pi(j) = \pi(i) + 1$, such that there is a node v in p whose query Q_v depends both on x_i and x_j is at least $\frac{1}{2}(n-1)$.

We first show (following the line of proof in [9]) that every permutation π has a half-complete path in T. To prove this, let $\pi = (1, 2, \ldots, n)$ (the identity permutation) for simplicity, and associate with each input $\mathbf{x}$ of S_π the set of all paths $\mathbf{x}$ can follow. This defines a coloring of the elements in S_π with at most $2^{n!}$ colors (if there are more than $n!$ paths the theorem follows immediately). By Ramsey's theorem, if $|S| \geq N(n, n+1, 2^{n!})$, then there is a set of $n+1$ elements $S_0 = \{a_1, \ldots, a_{n+1}\}$ (the a_i's are in increasing order) such that all the elements in $S_{0\pi}$ have the same color, that is, they all correspond to exactly the same set of paths. Denote this set of paths by P. Using Lemma 4.1, one can show that for each pair $(i, i+1)$ the probability that a query node that depends on both x_i and x_{i+1} occurs in a path in P must be at least $\frac{1}{2}$; otherwise, either $(a_1, \ldots, a_i, a_{i+1}, a_{i+3}, \ldots, a_{n+1})$ is accepted with probability $<\frac{3}{4}$ or $(a_1, \ldots, a_{i-1}, a_{i+1}, a_{i+1}, \ldots, a_{n+1})$ is rejected with probability $<\frac{3}{4}$. As a result, the expected number of distinct pairs $(i, i+1)$ in a path is at least $\frac{1}{2}(n-1)$, which implies that there exists at least one path with $\frac{1}{2}(n-1)$ distinct pairs.

Given a half-complete path p and its associated graph $G(p)$, we define a *half-Hamiltonian* path in $G(p)$ [9] as a Hamiltonian path in the complete graph K_n such that at least half of its edges are in $G(p)$. It is easy to see that, if p is half-complete for t permutations, then $G(p)$ contains t half-Hamiltonian paths. We now estimate the number of half-Hamiltonian paths in a graph G with n nodes and m edges.

With each sequence $B = (b_1, \ldots, b_{n-1})$, $b_i \in \{0, 1\}$, such that at least half of the b_i's are 1's, we associate a set H_B of all half-Hamiltonian paths in which the ith edge is in $G(p)$ iff $b_i = 1$. To bound the size of each such H_B, let $v_1, \ldots, v_n$ be an arbitrary path in H_B. Let d_i be the degree of v_i. There are n possibilities of choosing v_1. The number of ways to choose v_{i+1} is at most d_i if $b_i = 1$ and at most $n - i$, otherwise. Let $q \geq \frac{1}{2}(n-1)$ be the number of 1's in B, then

$$|H_B| \leq n \prod_{j=1}^{q} d_{i_j} \prod_{j=1}^{n-q} (n-j),$$

where $b_{i_1}, b_{i_2}, \ldots, b_{i_q}$ are the only 1's in B, and the d_{i_j}'s are the largest degrees in G. Obviously, $\sum_{j=1}^{q} d_{i_j} \leq 2m$. Under the constraints on the d_i's and q it is not hard to verify (assuming that $m < n^2/8$) that the right-hand side of the inequality above attains a maximum when all the d_{i_j}'s are equal and q is minimized. Thus,

$$|H_B| \leq n \left(\frac{2m}{n}\right)^{(1/2)n} \frac{(n-1)!}{\lceil \frac{1}{2}(n-1) \rceil !},$$

and since there are less than 2^n distinct such B's, the total number of half-Hamiltonian paths in G is less than

$$n 2^n \left(\frac{2m}{n}\right)^{(1/2)n} \frac{(n-1)!}{\lceil \frac{1}{2}(n-1) \rceil !}. \tag{4.2}$$

Now let T be a k-bounded probabilistic decision tree that recognizes EU, where $k \leq n^{1/2-\epsilon}$, and let h be the height of T. Then, for each computation path p in T, $G(p)$ contains at most $h(\frac{k}{2})$ edges. Let s be the number of distinct paths in T. There are $n!$ permutations; hence, we need to account for $n!$ half-Hamiltonian paths. Using (4.2), we get

$$s n 2^n \left(h \frac{k(k-1)}{n}\right)^{(1/2)n} \frac{(n-1)!}{\lceil \frac{1}{2}(n-1) \rceil !} \geq n!.$$

Rearranging terms, this yields

$$s n 2^n \left(h \frac{k(k-1)}{n}\right)^{(1/2)n} \geq n\lceil \tfrac{1}{2}(n-1)\rceil !.$$

Taking logarithms and rearranging terms again yields

$$\log s + n + \tfrac{1}{2}n(\log h + 2\log k - \log n) \geq \log(\lceil \tfrac{1}{2}(n-1)\rceil !) \geq (1-\epsilon)\tfrac{1}{2}n\log n \quad \text{(for large enough } n\text{)}.$$

Since $k \leq n^{(1/2)-\epsilon}$, $2\log k \leq (1-2\epsilon)\log n$. Thus,

$$\log s + n + \tfrac{1}{2}n\log h \geq (1+\epsilon)\tfrac{1}{2}n\log n.$$

If $\log s \geq \frac{1}{4}\epsilon n\log n - n$, then, since $h \geq \log s$, the theorem follows. Assume that $\log s \leq \frac{1}{4}\epsilon n\log n - n$; we get

$$n\log h \geq (1+\tfrac{1}{2}\epsilon)n\log n,$$

which implies that

$$\log h \geq (1+\tfrac{1}{2}\epsilon)\log n.$$

Thus, we actually get that, unless there are many leaves (namely, s is large enough), the height of the tree must satisfy $h = \Omega(n^{1+\epsilon})$. □

Note that the part of the proof using Ramsey's theorem can also be achieved by the technique of Section 3; however, this more direct application yields smaller constants.

5. *Conclusion and Further Research*

We have presented techniques for extending lower bound results for decision trees using simple comparisons to decision trees using general queries. The only restriction we impose on the queries is the number of inputs involved. In some cases we show that this restriction is also necessary. The techniques are purely combinatorial. As a result, the lower bounds apply to any large enough computational domain.

The first use of Ramsey's theorem we made here was inspired by a previous work of Yao [21]. The same technique has already been used to extend lower bounds proved for comparison-based algorithms to more general ones: Snir used it in [17] for parallel computations, and Frederikson and Lynch used it in [3] for distributed computations. All these applications of Ramsey's theorem share a common framework.

The result of Theorem 3.3 can be reformulated as follows (this formulation is also due to Ramsey [12]).

THEOREM 5.1. *For each j, k, m, n, there is a number $N(j, k, m, n)$ such that the following holds: Let F be a universal formula of size j in first-order predicate calculus, with predicates $P_1, \ldots, P_k$, and n variables. Then, if this formula can be satisfied by a model of size $N(j, k, m, n)$, it can be satisfied by a model of size m, where the predicates P_i are order invariant.*

Assume a computational model where the assertion "Algorithm A solves correctly an order-invariant problem Δ in t steps" can be formally expressed by a universal formula of first-order predicate calculus using predicates $<, P_1, \ldots, P_k$. The previous theorem implies that, if this formula is satisfied on a sufficiently large domain (with $<$ interpreted as a total order), then it is satisfied on a domain of size at least n, where $P_1, \ldots, P_k$ are order invariant. Thus, lower bounds proved for

"order-invariant algorithms" (e.g., algorithms that are represented by order-invariant predicates) are valid with no restriction on the predicates.

Our use of Ramsey's theorem and the other uses we mentioned all follow from that observation.

This general formulation also shows the limitations of this method: The claim is not valid if function symbols are used; thus, we can only model computations where each operation has a fixed number of possible outcomes (in the case of a decision tree—two). Also, the domain in which the problem is solved must be large enough for Ramsey's theorem to be applied.

The constraints of the k-bounded decision tree model can be weakened in several ways. We have explored in this paper one direction, namely, allowing the bound k to grow with the number of inputs. When $k = n$, the number of inputs, then the information-theoretic bound is correct. In general, one would like to establish trade-offs between the widths of the queries and the height of the decision tree. In this context, note that Theorem 4.6 can be extended to show that for every $r \in [0.5, 1]$ there is an n^r-bounded decision tree of height $O(n^{2-2r})$ that recognizes EU.

The number of distinct input values (i.e., the size of S) has to be very large, especially in the general case, owing to the repeated use of Ramsey's theorem. In fact, it seems that $\Omega(n \log n)$ steps are needed to solve EU, even when the number of distinct values is $O(n)$. Is it possible to avoid the use of Ramsey's theorem, and give a combinatorial proof of the $\Omega(n \log n)$ lower bound, when the domain has size $O(n)$?

The results of this paper can be interpreted as closure theorems, in the following sense. Given a problem that is defined using the order structure of its domain S, then an optimal solution exists that uses only the order structure; imposing additional structure (i.e., defining additional predicates) does not help. Note that the element-uniqueness problem is defined in terms of the equality relation. However, a decision tree that uses only tests for equality requires $\Omega(n^2)$ steps to solve EU. Adding a (arbitrary) total-order structure on S helps to solve the problem.

The same question, namely, finding a minimal extension of the structure for which a computational problem is defined, such that an optimal solution exists, can be raised for other structures. For example, can one show that if a problem is defined in R^n using polynomial inequalities of degree k, then an optimal solution exists that uses only comparisons with degree k polynomials? We conjecture this result to be true when the length of a path in the decision tree is defined to be the sum of the degrees of the polynomials occurring on it.

REFERENCES

1. BEN-OR, M. Lower bounds for algebraic computation trees. In *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing* (Boston, Mass., Apr. 25–27). ACM, New York, 1983, pp. 80–86.
2. DOBKIN, D. P., AND LIPTON, R. J. On the complexity of computations under varying sets of primitives. *J. Comput. Syst. Sci. 18* (1979), 86–91.
3. FREDERIKSON, G., AND LYNCH, N. A. The impact of synchronous communication on the problem of electing a leader in a ring. In *Proceedings of the 16th Symposium on Theory of Computing* (Washington, D.C., Apr. 30–May 2). ACM, New York, 1984, pp. 493–503.
4. FREDMAN, M. L. How good is the information theory bound in sorting? *Theor. Comput. Sci. 1* (1976), 355–361.
5. FREDMAN, M. L., AND WEIDE, B. On the Complexity of computing the measure of $\bigcup[a_i, b_i]$. *Commun. ACM 21*, 7 (July 1978), 540–544.
6. KUNG, H. T., LUCCIO, F., AND PREPARATA, F. P. On finding the maxima of a set of vectors. *J. ACM 22*, 4 (Oct. 1975), 469–476.
7. MAASS, W. On the use of inaccessible numbers and order indiscernibles in lower bound arguments for random access machines. In preparation.

8. MANBER, U. A probabilistic lower bound for checking disjointness of sets. *Inf. Proc. Lett. 19* (July 1984), 51–53.
9. MANBER, U., AND TOMPA, M. The complexity of problems on probabilistic, nondeterministic, and alternating decision trees. In *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing* (San Francisco, Calif., May 5–7). ACM, New York, 1982, pp. 234–244.
10. MANBER, U., AND TOMPA, M. The effect of number of Hamiltonian paths on the complexity of a vertex-coloring problem. *SIAM J. Comput. 13* (1984), 109–115.
11. RABIN, M. Proving simultaneous positivity of linear forms. *J. Comput. Syst. Sci. 6* (1972), 639–650.
12. RAMSEY, F. P. On a problem of formal logic. *Proc. London Math. Soc., 2nd ser. 30* (1930), 264–286.
13. REINGOLD, E. M. Computing the maxima and the median. In *Proceedings of the IEEE 12th Symposium on Switching and Automata Theory.* IEEE, New York, 1971, pp. 216–218.
14. REINGOLD, E. M. On the optimality of some set algorithms. *J. ACM 19*, 4 (Oct. 1972), 649–659.
15. SHAMOS, M. I. Geometric complexity. In *Proceedings of the 7th Annual ACM Symposium on the Theory of Computing* (Albuquerque, N.M., May 5–7). ACM, New York, 1975, pp. 224–233.
16. SNIR, M. Comparisons between linear functions can help. *Theor. Comput. Sci. 19* (1982), 321–330.
17. SNIR, M. On parallel searching. *Siam J. Comput.*, To appear.
18. SNIR, M. Lower bounds on probabilistic linear decision trees. *Theor. Comput. Sci.*, To appear.
19. STEELE, J. M., AND YAO, A. C. Lower bounds for algebraic decision trees. *J. Algorithms 3* (1982), 1–8.
20. YAO, A. C. On the complexity of comparison problems using linear functions. In *Proceedings of the 16th Symposium on Foundations of Computer Science.* IEEE, New York, 1975, pp. 85–89.
21. YAO, A. C.-C. Should tables be sorted? *J. ACM 28*, 3 (July 1981), 616–628.
22. YAO, A. C.-C. A lower bound to finding convex hulls. *J. ACM 28*, 4 (Oct. 1981), 780–789.

RECEIVED SEPTEMBER 1984; REVISED MARCH 1985; ACCEPTED MARCH 1985