APPLICATIONS OF RAMSEY'S THEOREM TO DECISION TREES COMPLEXITY

by

Shlomo Moran
Marc Snir
Udi Manber

Computer Sciences Technical Report #551

August 1984

APPLICATIONS OF RAMSEY'S THEOREM TO DECISION TREES COMPLEXITY

(Preliminary Version)

Shlomo Moran
Department of Computer Science
Technion, Haifa, Israel

Marc Snir
Department of Computer Science
Hebrew University, Jerusalem, Israel

Udi Manber*
Department of Computer Science
University of Wisconsin
Madison, WI 53706

ABSTRACT

Combinatorial techniques for extending lower bounds results for decision trees to general types of queries are presented. We consider problems, which we call *order invariant*, that are defined by simple inequalities between inputs. A decision tree is called $k-bounded$ if each query depends on at most k variables. We make no further assumptions on the type of queries. We prove that we 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.

We also prove an $\Omega(n \log n)$ lower bound for the element uniqueness problem and several other problems for any k-bounded decision tree, such that $k = 0(n^c)$ and $c < ½$. This lower bound is tight since that there exist $n^{½}$-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.

1. INTRODUCTION

Decision trees are very useful in proving lower bounds for combinatorial problems. In particular, they have been extensively used to analyze sorting-type problems whose outcome depends on the relative order of the inputs ([2], [5], [6], [9], [13], [14], [20]).

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 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 [12] extended the $n-1$ lower bound for maximum finding to decision trees using linear comparisons: Yao [19], and Dobkin and Lipton [2] did the same for the selection problem, and the element uniqueness problem, respectively. Rabin [10] 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 [18]). Manber and Tompa [8] extended several lower bounds to nondeterministic and probabilistic models of decision trees (see also [7] and [17]).

All these results assume that the inputs are taken from R, the set of real numbers. This allows the authors to use sophisticated geometrical 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. In 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

*Supported in part by the National Science Foundation under Grant MCS83-03134.

To appear in the 25th Annual Symposium on Foundation of Computer Science (October 1984).

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 non-constant arity. We use the combinatorial techniques developed in [8] 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 $c < ½$. This is a tight result in the sense that if $k = n^{½}$ 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 then 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 can be extended to nondeterministic decision trees and probabilistic decision trees using the techniques of [8] and [17].

2. DEFINITIONS

Let S be a totally ordered set and n a positive integer. Let S^n denotes the set of all n-tuples of elements of S, and let $[S]^n$ denotes 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 $\bar{x}$ and $\bar{y}$ are *order equivalent*, $\bar{x}\equiv\bar{y}$, if for each i and j $x_i < x_j \Longleftrightarrow y_i < y_j$. We call the equivalence class of $\bar{x}$ the *order type* of $\bar{x}$. If π is a permutation of n elements, then the *order type of* π *in* S, denoted 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 $\bar{x}$ proceeds downward from the root. If the node v is reached then the predicate Q_v is evaluated on $\bar{x}$, and one of the outgoing edges is chosen, according to the outcome of the evaluation. The path $\bar{x}$ follows is called the *computation path* for $\bar{x}$. The tree T *solves* Δ *on* C if, for each $\bar{x} \in C$, $\bar{x}$ reaches a leaf with label D_i iff $\bar{x} \in D_i$; T solves Δ if it solves it on S^n, the domain of the problem.

We shall consider, in particular, *recognition problems*, i.e. decision problems which 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 [8] 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 ½. 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 $\geq$ ½. 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 $\geq 3/4$ of reaching an accepting leaf and we also require that all the other inputs have probability $\geq 3/4$ of reaching a rejecting leaf.

A predicate Q is *order invariant* if its truth set is order invariant, i.e. $\bar{x}\equiv\bar{y} \Longrightarrow Q(\bar{x})$ *iff* $Q(\bar{y})$; Q *is order invariant on* C if $\bar{x},\bar{y} \in C$, $\bar{x}\equiv\bar{y} \Longrightarrow Q(\bar{x})$ *iff* $Q(\bar{y})$. The decision tree T *is order invariant on* C if each predicate occurring in T is of 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 [8] 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 one, 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. $Cr_k(\Delta) \le Cr_2(\Delta) \le O(k \log k) Cr_k(\Delta)$.

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 D.

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

□

We make use of the following well known theorem [11].

RAMSEY'S THEOREM: For any n, m and q there exist 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.

THEOREM 3.3. For each m, n and t there exist 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 exist a set $C \subset S$ such that $|C| \ge m$, and T is order invariant on C^n.

Proof: Let $Q_1, \ldots, Q_p$ be the distinct predicates labeling the nodes of T. Let $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_k\}$ be two k-element subsets of S, indexed in increasing order. We say that $\{x_1, \ldots, x_k\}$ is *congruent* to $\{y_1, \ldots, y_k\}$ if, for each mapping $\sigma: 1..n \to 1..k$, and each $1 \le j \le p$

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

It is easy to see that this is indeed an equivalence relation on $[S]^k$. The number G of equivalence classes of this relation is bounded by $2^{pn^k} < 2^{2^t n^k}$. 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 C such that $|C| \ge s$ and all elements of $[C]^k$ 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 $\bar{x}$ and $\bar{y}$ be two order equivalent tuples in C^n. Let $x'_1, \ldots, x'_k$ be the distinct components of $\bar{x}$, indexed in increasing order, and let $y'_1, \ldots, y'_k$ be similarly defined for $\bar{y}$. Let $\sigma: 1..n \to 1..k$ be such that $x_i = x'_{\sigma(i)}$, $i = 1, \ldots, n$. Since $\bar{x} \equiv \bar{y}$, it follows that $y_i = y'_{\sigma(i)}$, $i = 1, \ldots, n$. Since $\{x'_1, \ldots, x'_k\}$ is congruent to $\{y'_1, \ldots, y'_k\}$, $Q_j(\bar{x})$ *iff* $Q_j(\bar{y})$, for any predicate Q_j occurring in T. Thus, T is order invariant on C^n.

□

THEOREM 3.4. For each m, n, k and t there exist 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 Theorem 3.3, if S is large enough, then there exist a set C such that $|C| \ge n$ and T is order invariant when restricted to inputs from C^n. Each tuple $\bar{x} \in S^n$ is order equivalent to a tuple $\bar{y} \in C^n$ (since $|C| \ge n$). Replace each predicate Q occurring in T by the predicate Q' defined as follows: $Q'(\bar{x}) = Q(\bar{y})$, where $\bar{y} \in C^n$ is order equivalent to $\bar{x}$. By the previous remark such tuple $\bar{y}$ exists. As Q is order invariant on C^n, the definition does not depend on the choice of $\bar{y}$ 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 $\bar{x} \in C^n$, then $\bar{x}$ 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 3.5. 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 3.6. The results in Corollary 3.5 hold for probabilistic and nondeterministic decision trees as defined in [8].

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 non-constant 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 which 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 shall 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 which 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 $Q(s_1) = Q(s_2)$ 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 which 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 Theorem, there is a subset S_0 of S such that $|S_0| \ge n+1$ 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 $\pi = (1, 2, \ldots, n)$, 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 $1 \le i < n$. Then since both

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

and

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

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

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

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 s.t. if $n > n_\epsilon$ then $log(n!) > (1-\epsilon)n \log n$

The following definitions and Lemmas are similar to the ones in [8] and are outlined here. Given a computation path p, $G(p) = (V,E)$ is an undirected graph such that $V = \{1,\ldots,n\}$ and $E = \{(i, j)|$ there is a query in p that involves both x_i and $x_j\}$.

LEMMA 4.4. If a path p is complete for t permutations, Then $G(p)$ contains t Hamiltonian paths.

LEMMA 4.5. The number of Hamiltonian paths on graph on n vertices $(n > 1)$ and e edges is at most $n(e/(n-1))^{n-1}$.

THEOREM 4.6. There exists a function $N = N(\epsilon, n)$ such that any $n^{\frac{1}{2}-\epsilon}$-bounded decision tree T that recognizes EU on a set S such that $|S| \ge N$, has height $\Omega(n \log n)$.

Proof: Let T be a decision tree that solves EU. By Theorem 3.4 we can assume without loss of generality that T is order invariant. If T is k-bounded and of height h then for each p $G(p)$ contains at most $h\binom{k}{2}$ 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, $\chi \cdot t \ge n!$. Taking into account that $\chi \le 2^h$, and using Lemma 4.5 to bound t, we obtain

$$2^h n[\binom{k}{2}h/(n-1)]^{n-1} \ge n!$$

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

$$h + \log n + (n-1)[2log\, k + \log(h/(n-1))] > \log(n!) > (1-\epsilon)n \log n$$

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

$$h + (n-1)\log(h/(n-1)) > \epsilon(n-2)\log n,$$

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

□

THEOREM 4.7. There exist $n^{\frac{1}{2}}$-bounded deterministic decision trees of height $O(n)$ that solve the element uniqueness problem.

Proof: Divide the n elements into $2\lceil n^{\frac{1}{2}}\rceil$ blocks of size $\le \frac{1}{2}\lceil n^{\frac{1}{2}}\rceil$, and check for every pair of blocks whether their union in 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 4.8. The complexity of k-bounded deterministic decision trees, where $k \leq n^{\frac{1}{2}-\epsilon}$, 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 4.6 and will be omitted here. The reader is referred to [8] for more details.

□

4.2. Probabilistic decision trees

We now consider probabilistic decision trees and show that the results obtained in Theorem 4.6 hold for this model as well. The next theorem is an extension of Theorem 8 in [8], where the same lower bound was proven for probabilistic decision trees with only simple comparisons.

THEOREM 4.9. There exists a function $N = N(\epsilon, n)$ such that any $n^{\frac{1}{2}-\epsilon}$-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 $(n-1)/2$.

We first show (following the line of proof in [8]) that every permutation π has a half-complete path in T. To prove this let $\pi = (1,\ldots,n)$ for simplicity, and associate with each input x of S_π the set of all paths x can follow. This defines a coloring of the elements in S_π with at most $2^{n!}$ colors. 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_π} have the same color, i.e. 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 1/2; otherwise either $(a_1, \ldots a_i, a_{i+2}, \ldots a_{n+1})$ is accepted with probability $< 3/4$ or $(a_1, \ldots, a_{i-1}, a_{i+1}, a_{i+1}, \ldots, a_n)$ is rejected with probability $< 3/4$.

Given a half-complete path p and its associated graph $G(p)$, we define a *half-Hamiltonian* path in $G(p)$ ([8]) 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. The number of half-Hamiltonian paths in a graph with n vertices and e edges is shown in [8] to be at most

$$n 2^n \left(\frac{2e}{n}\right)^n \frac{(n-1)!}{(\frac{1}{2}n-1)!}$$

The lower bound can now be derived using arguments similar to Theorem 4.6.

□

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 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 proven for comparison based algorithms to more general ones: Snir used it in [15] for parallel computations, and Frederikson and Lynch used it in [3] for distributed computations. Thus, Ramsey's theorem seems to be a powerful tool for concrete computational complexity.

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 tradeoffs between the queries' widths and the height of the decision tree. In this context, note that Theorem 4.7 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, due 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 an (arbitrary) total order structure on S, helps to solve the problem.

The same question, namely finding a minimal extension of the structure where 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] M. Ben-Or, "Lower bounds for algebraic computation trees", Proc. 15th ACM Symp. on Theory of Computing (1983) 80-86.

[2] D. P. Dobkin and R. J. Lipton "On the Complexity of computations under Varying Sets of Primitives," J. Comput. System Sci. 18 (1979) 86-91.

[3] G. Frederikson and N. A. Lynch, "The impact of synchronous communication on the problem of electing a leader in a ring", Proc. 16th Symp. on Theory of Computing (1984) 540-544.

[4] M. L. Fredman, "How Good is the Information Theory Bound in Sorting?", Theoretical Computer Science 1 (1976) 355-361.

[5] M. L. Fredman and B. Weide, "On the Complexity of computing the Measure of $\bigcup[a_i, b_i]$," Comm. ACM 21 (1978) 540-544.

[6] H.T. Kung, F. Luccio, and F.P. Preparata, "On finding the maxima of a set of vectors". J. ACM 22 (1975) 469-476.

[7] U. Manber, "A Probabilistic Lower Bound for Checking Disjointness of Sets", To appear in Information Processing Letters

[8] U. Manber and M. Tompa, "The Complexity of Problems on Probabilistic, Nondeterministic, and Alternating Decision Trees," Proc. 14th ACM Symp. on Theory of Computing (1982) 234-244

[9] U. Manber and M. Tompa, "The Effect of Number of Hamiltonian Paths on the Complexity of a Vertex-Coloring Problem," SIAM Journal on Computing 13 (1984) 109-115.

[10] M. Rabin, "Proving simultaneous positivity of linear forms", J. Comput. System Sci. 6 (1972) 639-650.

[11] F. P. Ramsey, "On a problem of formal logic", Proc. London Math. Soc., 2nd ser, 30 (1930) 264-286.

[12] E.M. Reingold, "Computing the maxima and the median", Proc. IEEE 12th Symp. on Switching and Automata Theory (1971) 216-218.

[13] E.M. Reingold, "On the Optimality of Some Set Algorithms", Journal of the ACM, Volume 19 (1972), 649-659.

[14] M. I. Shamos, "Geometric Complexity," Proc. 7th ACM Symp. on Theory of Computing (1975) 224-233.

[15] M. Snir, "On parallel searching", to appear in Siam J. of Computing.

[16] M. Snir, "Comparisons between linear functions can help", Theoretical Computer Science, 19 (1982) 321-330.

[17] M. Snir, "Lower Bounds on Probabilistic Linear Decision Trees", to appear in Theoretical Computer Science.

[18] J. M. Steele and A. C. Yao, "Lower bounds for algebraic decision trees", J. of Algorithms 3 (1982) 1-8.

[19] A. C. Yao, "On the complexity of comparison problems using linear functions", Proc. 16th. Symp. on Foundations of Computer Science (1975) 85-89.

[20] A. C. Yao, "A lower bound to finding convex hulls", J. ACM 28 (1981) 780-789.

[21] A. C. Yao, "Should tables be sorted?" J. ACM 28 (1981) 615-628.