On Read-Once Threshold Formulae and their Randomized Decision Tree Complexity

(Preliminary Version)

Rafi Heiman * Ilan Newman † Avi Wigderson ‡

Abstract

TC^0 is the class of functions computable by polynomial-size, constant- depth formulae with threshold gates. Read-Once TC^0 (RO-TC^0) is the subclass of TC^0 which restricts every variable to occur exactly once in the formula.

Our main result is a (tight) linear lower bound on the randomized decision tree complexity of any function in RO-TC^0.

This relationship between threshold circuits and decision trees bears significance on both models of computation. Regarding decision trees, this is the first class of functions for which such a strong bound is known. Regarding threshold circuits, it may be considered as a possible first step towards proving $TC^0 \neq NC^1$; generalizing our lower bound to all functions in TC^0 will establish this separation.

Another structural result we obtain is that a read-once threshold formula uniquely represents the function it computes.

1 Introduction

1.1 Boolean Decision Trees

The Boolean decision tree is an extremely simple model for computing Boolean functions. It charges only for reading input variables. Every function on n variables has complexity $\leq n$. Perhaps surprisingly, decision trees turned out to be fundamental in studying the complexity of Boolean functions in general models, such as CREW PRAM [Nis89], and AC^0 circuits [LMN89].

The first major result for this model was the linear lower bound of Rivest and Viullemin [RV78] for the class of monotone graph properties, proving the Aanderaa-Rosenberg conjecture.

A conjecture that an $\Omega(n)$ lower bound applies to this class even if we allow randomization, is attributed to Karp. This has been proven for a few special monotone graph properties, but the best general lower bound is $\Omega(n^{2/3})$ of Hajnal [Haj88] (improving on Yao [Yao87] and King [Kin88]).

Our main result exhibits a natural class of functions for which a linear lower bound holds. The proof combines generalizing techniques developed in [SW86] to study read-once formulae, and understanding 'partial' computation of threshold functions by decision trees.

1.2 Threshold Circuits

The study of circuits with threshold gates and, in particular, those of polynomial size and constant depth (the class TC^0) also has several motivations. These circuits capture essential aspects in neural net computations [RMP86], [Hop82]. They have been shown to be equivalent to constant-depth arithmetic circuits over finite fields [Rei87], [SFB]. They were recently related to simulating the Polynomial hierarchy by counting oracles [Tod89], [All89].

The fundamental question of whether the inclusion $TC^0 \subseteq NC^1$ is proper, surfaced naturally after $AC^0 \neq TC^0$ was resolved ([FSS84],

*The Weizmann Institute of Science, Rehovot, Israel

†The Hebrew University, Jerusalem, Israel

‡The Hebrew University, Jerusalem, Israel. Partially supported by Israel-American Binational Science Foundation Grant No. 87-00082

CH2899-3/90/0000/0078$01.00 © 1990 IEEE

[Ajt83] and their improvements), and after the results about constant depth circuits with prime modulo gates were proved ([Raz87], [Smo87]). This question has been under attack in the last few years.

Two important steps were made in the direction of separating TC^0 from NC^1. The first, by Hajnal et al. [HMP*87], separated depth-2 from depth-3 polynomial-size threshold circuits. The second, by Yao [Yao89], separated the monotone analogues of the classes TC^0 and NC^1.

In 1986 Saks suggested a bold approach to separating these classes: Show that every function in TC^0 has high (say linear) randomized decision tree complexity (in terms of its deterministic complexity). This will suffice, as there are several examples ([Sni85], [Bop], [SW86]) of evasive (deterministic complexity n) functions in NC^1 with randomized complexity n^α for $\alpha < 1$.

This approach reduces a lower bound in the Circuits model to a lower bound in the information theoretical model of randomized decision trees. It is particularly original and intriguing, since the separation will be proved by showing that functions in the smaller class are harder (in the second model).

Our result can be considered as a first step in this direction. It proves the desired lower bound for read-once TC^0 functions. It is naive to be optimistic just because every TC^0 function is a simple projection of a read-once TC^0 function; it is not clear what happens to decision tree complexity under projections. However, the proof of the lower bound reveals that, from the point of view of randomized decision trees, threshold gates are no more powerful than ANDs and ORs, which hints that this may be the right direction to pursue.

2 Definitions and Statement of Results

2.1 Boolean Decision Trees

A *deterministic decision tree*, T, is a labeled binary tree. Each non-leaf node is labeled by some input-variable, x_i. The two outgoing edges of such nodes are labeled, one by '1' and the other by '0'. Each leaf is labeled by an output value which is either '1' or '0'.

The *path* of T on input-setting $\varepsilon = \varepsilon_1, ..., \varepsilon_n \in \{0,1\}^n$, $\mathrm{Path}_T(\varepsilon)$, is that (unique) path in the tree which starts at the root, and at each node, say labeled by x_i, follows the edge labeled ε_i. $\mathrm{Var}_T(\varepsilon)$ denotes the set of variables labeling the nodes of $\mathrm{Path}_T(\varepsilon)$. The *output* of T given ε, $\mathrm{Output}_T(\varepsilon)$, is the bit labeling the leaf of $\mathrm{Path}_T(\varepsilon)$. T *computes* the Boolean function f if $\mathrm{Output}_T(\varepsilon) = f(\varepsilon)$ for every ε.

The *time* consumed by T, $\mathrm{Time}_T(\varepsilon)$, is simply $|\mathrm{Var}_T(\varepsilon)|$. (Every variable is probed at most once in a path.) The *complexity* of T is the time consumed for a worst case input. The *deterministic decision tree complexity* of f, $DC(f)$, is the complexity of the best deterministic decision tree that computes f,

$$DC(f) = \min_T \max_\varepsilon \mathrm{Time}_T(\varepsilon). \tag{1}$$

A *randomized decision tree* for f, RT, is a distribution over the deterministic decision trees for f. Given ε, a deterministic decision tree is chosen according to this distribution and 'performed'. This makes the path and the time consumed random variables (however the output is always correct). The complexity of RT is the expected time (the expected number of variables it probes in order to determine the output) for a worst case input. The *randomized decision tree complexity* of f, $RC(f)$, is the complexity of the best randomized decision tree that computes f,

$$RC(f) = \min_{RT} \max_\varepsilon \mathrm{E}_{T\in RT}[\mathrm{Time}_T(\varepsilon)]. \tag{2}$$

Here E stands for expectation and $T \in RT$ stands for a random T chosen according to the distribution RT.

By a lemma of Yao [Yao77], which is based on the minimax theorem, we have the following equivalence between $RC(f)$ and the *distributional complexity* of f.

$$RC(f) = \max_D \min_T \mathrm{E}_{\varepsilon\in D}[\mathrm{Time}_T(\varepsilon)]. \tag{3}$$

where D ranges over all distributions on input settings to f, T ranges over all deterministic decision trees for f, and $\varepsilon \in D$ stands for a random input-setting ε chosen according to the distribution D. The distributional complexity is a useful tool for proving lower bounds. One can guess some D and then prove a lower bound on $\min_T \mathrm{E}_{\varepsilon \in D}[\mathrm{Time}_T(\varepsilon)]$.

A *partial decision tree*, T, for f is very similar to a deterministic one, except that a leaf in it may contain a '?'. T is required to satisfy $\mathrm{Output}_T(\varepsilon) = f(\varepsilon)$ for every ε with $\mathrm{Output}_T(\varepsilon) \neq$ '?'. For example, the trivial decision tree, which contains a single node (a leaf) labeled by a '?', is a partial decision tree for every Boolean function. Central to our proof is an inequality satisfied by all partial decision trees computing a simple threshold function.

2.2 Read-Once Threshold Formulae

A *threshold gate*, denoted T_l^k for some $k > 1$ and $1 \leq l \leq k$, is a Boolean gate with k inputs that outputs '1' iff at least l of its inputs are '1'. For example, T_1^k and T_k^k are, respectively, OR and AND gates of fan-in k.

A *read-once threshold formula* is a formula with threshold gates in which each variable appears exactly once. We would like to point out here that disallowing negation gates doesn't restrict the generality of our results. Negation gates can be 'pushed' to be applied to inputs only. Then renaming all negative literals as positive ones (as input-variables) doesn't change relevant combinatorial properties such as the deterministic and the randomized decision tree complexities.

An example of read-once AND-OR formula is the AND-OR tree function, $g^{(d)}$, defined for every depth d on $n = 2^d$ input-variables:

$$g^{(0)}(x_1) = x_1; \quad \text{and}$$
$$g^{(d+1)}(x_1, \ldots, x_{2^{d+1}}) =$$
$$g^{(d)}(x_1, \ldots, x_{2^d}) \Diamond g^{(d)}(x_{2^d+1}, \ldots, x_{2^{d+1}}).$$

where $\Diamond$ is AND for even d and is OR for odd d. This function is in NC^1; its formula depth is logarithmic in the number of variables. It is easy to see that its deterministic decision tree complexity is maximal, $DC(g^{(d)}) = n$. However its randomized complexity is low, $RC(g^{(d)}) = \Theta(n^\alpha)$ for $\alpha = \log_2(\frac{1+\sqrt{33}}{4}) = 0.753...$ [SW86]. The large (logarithmic) depth enables iterated savings that turn out to give this low randomized complexity.

2.3 Statement of Results

Our main result says that large depth is necessary for low randomized complexity.

Theorem 1: Let F be a read-once threshold formula of depth d over n input-variables that computes a Boolean function f. Then

$$RC(f) \geq \frac{n}{2^d}$$

The next section is devoted to the proof of this theorem. The proof is based on generalizing techniques of [SW86], as well as on using the new concept of partial decision trees. A weaker lower bound, namely $RC(f) \geq \frac{n}{4^d}$, can be proven more simply by using the lower bound result of [SW86]. The direct proof given here is, we believe, a more significant step in the study of the randomized decision tree complexity in general, and that of threshold circuits in particular. This direct proof has another advantage. It works also in a more powerful model. This model enables, in particular, gates that compute arbitrary symmetric functions:

Definition: A Boolean function g, defined on k input-variables, is said to *contain a flip* if there exists an l, $1 \leq l \leq k$, such that g outputs the same value whenever exactly l of its inputs are '1', and outputs the opposite value whenever exactly $l-1$ of its inputs are '1'.

Corollary (of the proof): Let F be a read-once formula of depth d over n input-variables whose gates are functions that each contains a flip. Let f be the function F computes. Then

$$RC(f) \geq \frac{n}{2^d}$$

One may verify that the proof given in the next section works for these gates as well.

Our second result says that a Boolean function that can be represented by a read-once threshold formula has a unique such representation.

Definition: A read-once threshold formula is *non-degenerate* if no input of some T_1^k-gate (OR) is the output of some other $T_1^{k'}$-gate, and similarly, no input of a T_k^k-gate (AND) is the output of a $T_{k'}^{k'}$-gate.

Theorem 2: Two non-degenerate read-once threshold formulae that compute the same Boolean function are identical.

This theorem is proved in section 4.

3 Proof of Theorem 1

In the definitions of time and complexity above we assumed a unit cost for probing a variable. In order to carry an induction argument, we generalize these notions, and define them relative to a *variables cost function*, $c : \{x_1, ..., x_n\} \to \mathbf{R}$. Given such c we define $\mathrm{Time}_{c,T}(\varepsilon) = \sum_{x_i \in \mathrm{Path}_T(\varepsilon)} c(x_i)$. $DC(f,c)$ and $RC(f,c)$ denote the complexities relative to c and are defined similar to (1) and (2). (3) becomes then

$$RC(f,c) = \max_D \min_T \mathrm{E}_{\varepsilon \in D}[\mathrm{Time}_{c,T}(\varepsilon)]. \quad (4)$$

3.1 Overview of the Proof

For a formula consisting of a single threshold gate the proof is not very difficult, even if variables have non-unit costs. One can use this case as a single step in a top-bottom inductional proof. However this doesn't yield a lower bound on $RC(f)$, rather a lower bound on the complexity of *directional* randomized decision trees for f. Directionality means that variables are probed in a restricted manner, depending on the formula's structure; if any variable in any sub-formula is probed then after this probe the decision tree must figure out first the value of that sub-formula before probing any variable that appears in another part of the formula.

That is the reason for the need of a bottom-up induction given in the next sub-section. The bottom-up method forces the single step of that induction (the shrinking lemma) to make a global statement on the formula. In the lemma's proof (sub-section 3.3) we carefully define a distribution D on inputs and a (set of) decision tree(s) T', that enable reducing the lemma's statement into a statement involving a simple threshold formula only (i.e., a single gate). The analogue to the evaluation of a simple threshold function (for the directional case) is a claim on partial decision trees that compute a simple threshold function (for the general case). Sub-section 3.4 is devoted to this claim.

3.2 Reducing Theorem 1 to the Shrinking Lemma

The shrinking lemma: Let F be a read-once threshold formula of depth $d > 0$ that computes a Boolean function f. Consider an internal gate T_l^k whose entries are all variables. Denote these variables by $Y = \{y_1, ..., y_k\}$. Denote the rest of the variables $X = \{x_1, ..., x_m\}$. (See figure 1.) Let $c : X \cup Y \to \mathrm{R}$ be a cost function for the $m + k$ variables of f. Let F' be the formula obtained from F by replacing the sub-formula $T_l^k(y_1, ..., y_k)$ by a single variable v (see figure 2), and let f' be the function computed by F'. Define a new cost function, c', by $c'(x_i) = c(x_i)\ \forall 1 \le i \le m$ and $c'(v) = \frac{c(Y)}{2}$ where $c(Y) = \sum_{i=1}^{k} c(y_i)$.
Then $RC(f', c') \le RC(f, c)$.

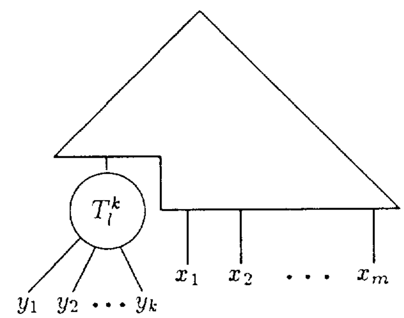

Figure 1: The given F

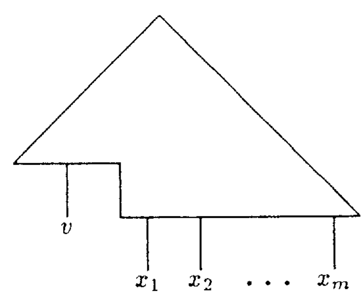

Figure 2: The shrunk F'

Theorem 1 follows by applying the lemma inductively. The beginning is with unit variables cost. The last shrinking yields the simple formula consisting of a single variable, v', whose cost bounds $RC(f)$ from below, and is

$$c(v') = \sum_{i=1}^{n} 2^{-\mathrm{Depth}(x_i)} \ge \frac{n}{2^d}$$

Here $\mathrm{Depth}(x_i)$ and d relate, respectively, to the depth of a variable x_i (which is well defined since x_i appears only once) and to the maximal depth over all variables in the (original) formula F.

3.3 Reducing the Shrinking Lemma to the Claim

First, we introduce some necessary notations.

Notations:

$[k]$ denotes the set $\{1, ..., k\}$.

$\binom{A}{a}$ denotes the set of all subsets of a set A of cardinality a.

θ^1 (resp. θ^0) denotes the extension of a (partial) setting $\theta : X \rightarrow \{0,1\}$, on $X \cup \{v\}$ by $\theta^1(v) = 1$ (resp. $\theta^0(v) = 0$).

θ_M denotes the extension of $\theta : X \rightarrow \{0,1\}$ on $X \cup Y$ by $\theta_M(y_i) = 1_{i \in M}$ (i.e., 1 if $i \in M$ and 0 otherwise) where $M \subseteq [k]$.

$\Pr_D(E)$ denotes the probability of E given a distribution D.

$c(U)$ denotes the total cost of a subset U of input variables, $c(U) = \sum_{u \in U} c(u)$.

U_ε^T denotes the variables in some subset U of inputs that are probed by T given an input-setting ε) (to all variables), $U_\varepsilon^T = U \cap \mathrm{Var}_T(\varepsilon)$.

For proving the shrinking lemma we have to show $RC(f', c') \leq RC(f, c)$. Using (4) we show that
$\forall$ distribution D' on the input-settings to f'
$\exists$ distribution D on the input-settings to f s.t.
$\forall$ deterministic decision tree T for f
$\exists$ deterministic decision tree T' for f' with

$$\mathrm{E}_{\varepsilon' \in D'}[\mathrm{Time}_{c',T'}(\varepsilon')] \leq \mathrm{E}_{\varepsilon \in D}[\mathrm{Time}_{c,T}(\varepsilon)]. \tag{5}$$

So let D' be given. Define a distribution D as follows. For every X-setting $\theta : X \rightarrow \{0,1\}$ and a subset $M \subseteq [k]$ define

$$\Pr_D(\theta_M) = \begin{cases} \Pr_{D'}(\theta^1) \cdot \Pr(M) & \text{if } |M| = l \\ \Pr_{D'}(\theta^0) \cdot \Pr([k] \setminus M) & \text{if } |M| = l - 1 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where

$$\Pr(S) = \frac{\sum_{i \in S} c(y_i)}{\binom{k-1}{|S|-1} \cdot c(Y)} \tag{7}$$

for a non-empty set $S \subseteq [k]$. (The point here is to split $\Pr_{D'}(\theta^1)$ and $\Pr_{D'}(\theta^0)$ among the difficult to separate extensions of θ. These are the extensions θ_M for which $|M| = l$ or $|M| = l - 1$. The 'piece' of probability that such extension gets is proportional to the cost of the 'meaningful' Y-variables in it.)

Now, let T be given. We don't define T' explicitly, rather we define a set of candidate deterministic decision trees and prove that (5) holds for (at least) one of them. The candidates are the following $k \cdot \binom{k-1}{l-1}$ decision trees, $T_{(i,W)}$, indexed by pairs (i, W) where $i \in [k]$ and $W \in \binom{[k] \setminus \{i\}}{l-1}$.

$T_{(i,W)}$ is defined as the 'projection' of T under the following actions:

1. Each question 'y_i?' (in T) is replaced by the question 'v?' (in $T_{(i,W)}$).

2. For each $j \in W$, $T_{(i,W)}$ assumes that $y_j = 1$. Namely, for each node of T containing the question 'y_j?', $T_{(i,W)}$ bypasses this question to the 1 direction while deleting that node and the whole sub-tree under the 0 direction. (See figure 3.)

3. For each other j (i.e., $j \in [k] \setminus W$, $j \neq i$), $T_{(i,W)}$ assumes that $y_j = 0$: for each node of T containing 'y_j?', $T_{(i,W)}$ similarly bypasses the question, this case to the 0 direction. (See figure 4.)

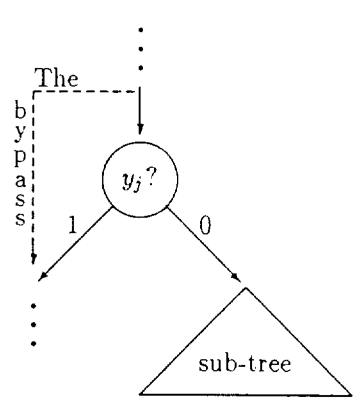

Figure 3: Assuming $y_j = 1$

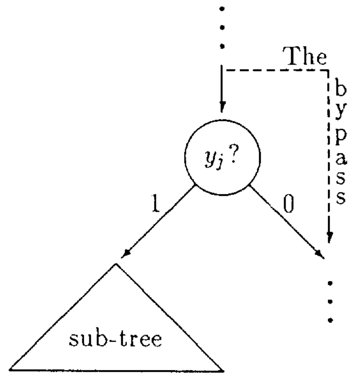

Figure 4: Assuming $y_j = 0$

Our remaining problem is to show that inequality (5) holds for some $T_{(i,W)}$. We show it by proving that the following convex combination of these $k \cdot \binom{k-1}{l-1}$ inequalities holds.

$$\sum_{i\in[k],W\in\binom{[k]\setminus\{i\}}{l-1}} p_{(i,W)} \mathrm{E}_{\varepsilon'\in D'}[\mathrm{Time}_{c',T_{(i,W)}}(\varepsilon')] \le \mathrm{E}_{\varepsilon\in D}[\mathrm{Time}_{c,T}(\varepsilon)], \qquad (8)$$

where the appropriate coefficients $\{p_{(i,W)}\}$ will be defined when used.

First we write the explicit terms for the two expectations above:

$$\begin{aligned} &\mathrm{E}_{\varepsilon'\in D'}[\mathrm{Time}_{c',T_{(i,W)}}(\varepsilon')] \\ \stackrel{\mathrm{def}}{=}\ & \sum_{\theta:X\to\{0,1\}} [\mathrm{Pr}_{D'}(\theta^1)\cdot \mathrm{Time}_{c',T_{(i,W)}}(\theta^1) \\ &+ \mathrm{Pr}_{D'}(\theta^0)\cdot \mathrm{Time}_{c',T_{(i,W)}}(\theta^0)], \end{aligned}$$

and

$$\begin{aligned} &\mathrm{E}_{\varepsilon\in D}[\mathrm{Time}_{c,T}(\varepsilon)] \\ \stackrel{\mathrm{def}}{=}\ & \sum_{\theta:X\to\{0,1\}} \sum_{M\subseteq[k]} \mathrm{Pr}_D(\theta_M)\cdot \mathrm{Time}_{c,T}(\theta_M)] \\ \stackrel{(6)}{=}\ & \sum_{\theta:X\to\{0,1\}} [\mathrm{Pr}_{D'}(\theta^1)\cdot \sum_{M\in\binom{[k]}{l}} \mathrm{Pr}(M)\cdot \mathrm{Time}_{c,T}(\theta_M) \\ &+ \mathrm{Pr}_{D'}(\theta^0)\cdot \sum_{M\in\binom{[k]}{l-1}} \mathrm{Pr}([k]\setminus M)\cdot \mathrm{Time}_{c,T}(\theta_M)]. \end{aligned}$$

Plugging these terms in (8) we note (due to the 'split'-manner definition of D) that it is sufficient to show for each $\theta : X \to \{0,1\}$ that

$$\sum_{i\in[k],W\in\binom{[k]\setminus\{i\}}{l-1}} p_{(i,W)}\mathrm{Time}_{c',T_{(i,W)}}(\theta^1) \le \sum_{M\in\binom{[k]}{l}} \mathrm{Pr}(M)\cdot \mathrm{Time}_{c,T}(\theta_M) \qquad (9)$$

and

$$\sum_{i\in[k],W\in\binom{[k]\setminus\{i\}}{l-1}} p_{(i,W)}\mathrm{Time}_{c',T_{(i,W)}}(\theta^0) \le \sum_{M\in\binom{[k]}{l-1}} \mathrm{Pr}([k]\setminus M)\cdot \mathrm{Time}_{c,T}(\theta_M) \qquad (10)$$

hold. By duality, we show only one of them, say (9). (9) implies (10) by changing the roles of 1-s and 0-s in the Y variables and considering the threshold gate T^k_{k-l+1}.

Separating the time to the costs of X and v, and to the costs of X and Y, and using the notations above, (9) is equivalent to

$$\sum_{(i,W)} p_{(i,W)}[c'(X^{T_{(i,W)}}_{\theta^1}) + c'(v)\cdot 1_{v\in \mathrm{Var}_{T_{(i,W)}}(\theta^1)}] \le \sum_{M\in\binom{[k]}{l}} \mathrm{Pr}(M)[c(X^T_{\theta_M}) + c(Y^T_{\theta_M})]. \qquad (11)$$

The key observation here is that $\mathrm{Path}_{T_{(i,W)}}(\theta^1)$ is the 'projection' of $\mathrm{Path}_T(\theta_{\{i\}\cup W})$ under actions 1–3 above. In particular, $X^{T_{(i,W)}}_{\theta^1} = X^T_{\theta_{\{i\}\cup W}}$ and $v \in \mathrm{Var}_{T_{(i,W)}}(\theta^1)$ iff $y_i \in Y^T_{\theta_{\{i\}\cup W}}$. Using these and (7), and enumerating the pairs (i,W) as $\{(M,i) : M \in \binom{[k]}{l},\ i \in M\}$, (11) is equivalent to

$$\sum_{M\in\binom{[k]}{l}} \sum_{i\in M} p_{(i,W)}\cdot [c'(X^T_{\theta_M}) + c'(v)\cdot 1_{y_i\in Y^T_{\theta_M}}]$$

$$\le \frac{1}{\binom{k-1}{l-1}} \sum_{M\in\binom{[k]}{l}} \sum_{i\in M} \frac{c(y_i)}{c(Y)}\cdot [c(X^T_{\theta_M}) + c(Y^T_{\theta_M})].$$

By definition, $c'(X^T_{\theta_M}) = c(X^T_{\theta_M})$. To cancel them we now define $p_i = \frac{c(y_i)}{c(Y)}$ and $p_{(i,W)} = \frac{p_i}{\binom{k-1}{l-1}}$ for $i \in [k]$ and $W \in \binom{[k]\setminus\{i\}}{l-1}$. (Note these coefficients are non-negative and their sum is 1.) So, by canceling and multiplying both sides by $\binom{k-1}{l-1}\cdot c(Y)$, the last inequality reduces to

$$c'(v)\cdot \sum_{M\in\binom{[k]}{l}} \sum_{i\in M} c(y_i)\cdot 1_{y_i\in Y^T_{\theta_M}}$$

$$\leq \sum_{M\in\binom{[k]}{l}} \sum_{i\in M} c(y_i)\cdot c(Y^T_{\theta_M}),$$

or, using the notation $Y_M = \{y_i \ : \ i \in M\}$,

$$c'(v)\cdot \sum_{M\in\binom{[k]}{l}} c(Y_M \cap Y^T_{\theta_M}) \leq \sum_{M\in\binom{[k]}{l}} c(Y_M)\cdot c(Y^T_{\theta_M}). \tag{12}$$

Note that we are left now with a problem involving the simple threshold sub-formula $F_{\text{sub}} = T^k_l(y_1, ..., y_k)$. The only role θ plays in (12) is to determine some projection of T that becomes a partial decision tree for F_{sub}. This projection is derived from T by bypassing each x_i?-question to the direction $\theta(x_i)$. It is partial since T may compute F without computing F_{sub}. In other words, the claim in the next subsection implies (12) and completes the whole proof.

3.4 The Reduced Claim

In the following claim a set $M \subseteq [k]$ is identified also with the input-setting to Y in which $y_i = 1$ iff $i \in M$.

The Partial Decision Tree Claim: Let T be a partial decision tree for $T^k_l(y_1, ..., y_k)$, and let c be a leaf cost function on Y. Then

$$\frac{c(Y)}{2}\cdot \sum_{M\in\binom{[k]}{l}} c(Y_M \cap Y^T_M) \leq \sum_{M\in\binom{[k]}{l}} c(Y_M)\cdot c(Y^T_M).$$

Proof of claim: The proof is by induction on k, with two base cases for each k.

Base case 1: $l = k$ **(AND gate).** The only $M \in \binom{[k]}{k}$ is $M = [k]$. Hence $Y_{[k]} = Y$ and the case follows.

Base case 2: $l = 1$ **(OR gate).** T doesn't probe a variable more than once, hence it is of the form of figure 5, where $0 \leq s \leq k$ and $Z = \{z_1, ..., z_s\} \subseteq Y$.

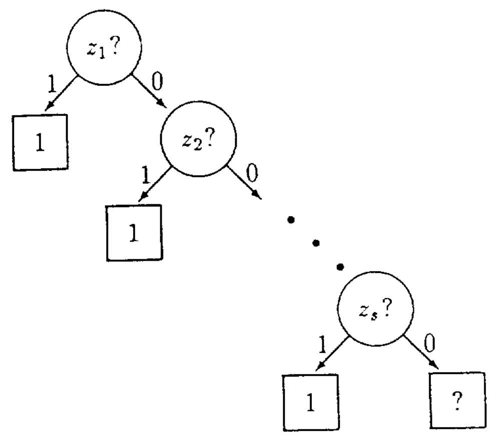

Figure 5: A partial decision-tree for OR

Denote $Y \setminus Z$ by $W = \{w_1, ..., w_{k-s}\}$. $M \in \binom{[k]}{1}$ encodes some element $Y_M \in Y$. If Y_M is some z_i then $Y^T_M = \{z_1, ..., z_i\}$, and if Y_M is some w_i then $Y^T_M = Z$. So what we have to show is

$$\frac{c(Y)}{2}\cdot\sum_{i=1}^{s} c(z_i) \quad \leq$$

$$\sum_{i=1}^{s}[c(z_i)\cdot \sum_{j=1}^{i} c(z_j)] + [\sum_{i=1}^{k-s} c(w_i)]\cdot[\sum_{j=1}^{s} c(z_j)].$$

This holds since the quantity

$$\frac{1}{2}[\sum_{i=1}^{s} c(z_i)]^2 + [\sum_{i=1}^{k-s} c(w_i)]\cdot[\sum_{i=1}^{s} c(z_i)]$$

is inbetween the inequality's two sides, due to

$$\frac{1}{2}[\sum_{i=1}^{s} a_i]^2 \leq \sum_{i=1}^{s}[a_i\cdot\sum_{j=1}^{i} a_j].$$

The induction step: $1 < l < k$ **(non trivial threshold gate).**

If T is trivial (doesn't probe any variable), then for every M, Y^T_M is empty, and the claim trivially holds.

Otherwise, let 'y_t?' be the first question of T, and let T_1 and T_0 be the subtrees under the directions $y_t = 1$ and $y_t = 0$ respectively (see figure 6).

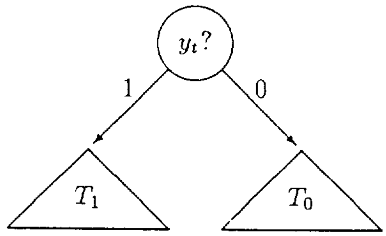

Figure 6: a non-trivial tree, T

For $M \ni t$, say $M = \{t\} \cup M'$, we have $Y_M = \{y_t\} \cup Y_{M'}$ and $Y_M^T = \{y_t\} \cup Y_{M'}^{T_1}$.
For $M \not\ni t$ we have $Y_M \subseteq Y \setminus \{y_t\}$ and $Y_M^T = \{y_t\} \cup Y_{M'}^{T_0}$.
In these terms the claim states that

$$\frac{c(Y)}{2} \cdot \{ \sum_{M' \in \binom{[k]\setminus\{t\}}{l-1}} [c(y_t) + c(Y_{M'} \cap Y_{M'}^{T_1})]$$

$$+ \sum_{M \in \binom{[k]\setminus\{t\}}{l}} c(Y_{M'} \cap Y_{M'}^{T_1})\}$$

$$\leq \sum_{M' \in \binom{[k]\setminus\{t\}}{l-1}} [c(y_t) + c(Y_{M'})] \cdot [c(y_t) + c(Y_{M'}^{T_1})]$$

$$+ \sum_{M \in \binom{[k]\setminus\{t\}}{l}} c(Y_M) \cdot [c(y_t) + c(Y_M^{T_0})].$$

T_1 and T_0 are partial decision-trees for $T_{l-1}^{k-1}(Y \setminus \{y_t\})$ and for $T_l^{k-1}(Y \setminus \{y_t\})$ respectively. So by induction,

$$\frac{c(Y \setminus \{y_t\})}{2} \cdot \sum_{M' \in \binom{[k]\setminus\{t\}}{l-1}} c(Y_{M'} \cap Y_{M'}^{T_1})$$

$$\leq \sum_{M' \in \binom{[k]\setminus\{t\}}{l-1}} c(Y_{M'}) \cdot c(Y_{M'}^{T_1}),$$

and

$$\frac{c(Y \setminus \{y_t\})}{2} \cdot \sum_{M \in \binom{[k]\setminus\{t\}}{l}} c(Y_M \cap Y_M^{T_0})$$

$$\leq \sum_{M \in \binom{[k]\setminus\{t\}}{l}} c(Y_M) \cdot c(Y_M^{T_0}).$$

Using these, and dividing by $c(y_t)$, the claim is reduced to

$$\frac{c(Y)}{2} \cdot \binom{k-1}{l-1} + \frac{1}{2} \sum_{M'} c(Y_{M'} \cap Y_{M'}^{T_1})$$

$$+ \frac{1}{2} \sum_{M} c(Y_M \cap Y_M^{T_0})$$

$$\leq \binom{k-1}{l-1} \cdot c(y_t) + \sum_{M'} c(Y_{M'})$$

$$+ \sum_{M'} c(Y_{M'}^{T_1}) + \sum_{M} c(Y_M),$$

and this holds due to

$$\frac{c(Y)}{2} \cdot \binom{k-1}{l-1} = \frac{1}{2} \sum_{M \in \binom{[k]}{l}} c(Y_M)$$

$$= \frac{1}{2}\binom{k-1}{l-1} \cdot c(y_t) + \frac{1}{2} \sum_{M' \in \binom{[k]\setminus\{t\}}{l-1}} c(Y_{M'})$$

$$+ \frac{1}{2} \sum_{M \in \binom{[k]\setminus\{t\}}{l}} c(Y_M).$$

This completes the proofs of the claim, the lemma and theorem 1.

4 Proof of Theorem 2

The proof is by induction on the number of variables n. The case of $n = 1$ is trivial.

Let f be computed by the two non-degenerate read-once threshold formulae $F_1 = T_l^k(h_1, \ldots, h_k)$ and $F_2 = T_s^r(g_1, \ldots, g_r)$. Since F_1 and F_2 are read once, each variable appears in a positive form (with no negation) in F_1 if and only if it appears in a positive form in F_2. Thus, we will assume from now on that F_1 and F_2 are monotone. (Change names of negative variables if there are any.)

The proof uses partial assignments and examines the restricted function and the restricted formulae. We note here that a restricted formula may be degenerate, however, in such a case we always change it to nondegenerate form by merging AND (OR) gates together and this does not change the type of the output gate.

Let H_i, $1 \leq i \leq k$ and G_j, $1 \leq j \leq r$ be the variable sets of h_i and g_j respectively.

Proposition 4.1 *If $H_i = G_j$ for some i, j then $h_i = g_j$ (as functions and as formulae).*

Proof: Any assignment to H_i that makes h_i to be '0' makes g_j to be '0' too, since it leaves the restricted function (looking on F_1) independent of the variables of H_i. But, if g_j does not become '0', the restricted function depends on at least one variable of G_j (looking on F_2). By the same argument on g_j we get that $h_i = g_j$ as functions. By the induction hypothesis they are identical as formulae, too. □

Proposition 4.2 *If* $1 < l < k$, $1 < s < r$ *and* $h_i = g_j$ *for some* i, j *then* F_1 *is identical to* F_2

Proof: Assume (w.l.o.g) that $i = j = 1$. If $l \geq 3$, assign '1' to the variables in H_1. F_1 reduces to $F_1' = T_{l-1}^{k-1}(h_2, ..., h_k)$ (where the output gate is not AND nor OR). By the induction hypothesis, F_2 reduces to the same formula. It follows that F_1 and F_2 are identical. Dually, if $l \leq k-2$ then by the assignment of the variables in H_1 to '0', we get the result. Therefore, we may assume that $l = s = 2$ and $k = r = 3$.

Assign '0' to the variables in H_2. F_1 reduces to $AND(h_1, h_3)$. By the induction hypothesis at least one of g_2, g_3 must become '0' (so that the restricted F_2 will also have AND as its output gate). Assume g_2 becomes '0'. It follows that $G_2 \subseteq H_2$. Now re-assign '0' to the variables in G_2. The same argument yields that $H_2 \subseteq G_2$. We have $H_2 = G_2$ and proposition 4.1 implies that $h_2 = g_2$. Similarly $h_3 = g_3$. □

We return to the proof of the theorem. Assume (w.l.o.g) that $H_k \cap G_r \neq \phi$. Let $x \in H_k \cap G_r$.

There are basicly two cases.

1. $1 < l < k$ and $1 < s < r$.

   If $h_k = g_r = x$ then by proposition 4.2 we are done. Otherwise, there is an assignment to x such that at least one of h_k and g_r does not become constant, say it is h_k. The output gate of F_1 does not change by this restriction (so it is not AND nor OR). By the induction hypothesis, the two restricted formulae must be identical. In particular, the output gate of F_2 doesn't become AND nor OR and since $k, r \geq 3$, there exist i, j, $i \neq k$, $j \neq r$ for which $h_i = g_j$. Again, by proposition 4.2 we are done.

2. $l = k$, i.e., $F_1 = AND(h_1, .., h_k)$.

   First assume that $s < r$ and get a contradiction as follows. Assign '1' to x. F_1 reduces to some non-constant formula, F_1'. F_2 reduces to either $T_s^r(g_1, ..., g_{r-1}, g_r')$ or $T_{s-1}^{r-1}(g_1, ..., g_{r-1})$ (but the latter is possible only if $s \geq 2$). In any case the output gate is not AND, and by the induction hypothesis so is for F_1'. This is possible only if $k = 2$ and $F_1' = h_1$. Comparing the variables sets of the two restricted formulae, we deduce that $H_2 \subseteq G_r$. We get the contradiction by assigning '0' to all variables of H_2; F_1 becomes '0' while F_2 doesn't.

   So far we got that $F_2 = AND(g_1, .., g_r)$. Assign '1' to the variables of H_k, getting $\text{AND}(h_1, ..., h_{k-1}) = \text{AND}(g_1', ..., g_r')$ (the latter might be degenerate). By the induction hypothesis on the h_i's and by the fact that each h_i can not have AND as its output gate (otherwise F_1 is degenerate), we get that for every $i \leq k-1$ there is some j such that $H_i \subseteq G_j$. Similarly, for every $j \leq r-1$ there is some i such that $G_j \subseteq H_i$. Note that the H_i's are pairwise disjoint, as well as the G_j's. It follows that $r = k$ and that for every $i < k$ there is a (unique) $j < r$ such that $H_i = G_j$. Therefore $H_k = G_r$, too. By proposition 4.1, $h_i = g_j$ for every pair i, j as above and also $h_k = g_r$.

The case where one of the output gates is OR gate, is dual to the last case above. □

Acknowledgements

We thank Uriel Feige, David Harel, Mike Saks and Moshe Tennenholz for helpful discussions and comments.

References

[Ajt83] M. Ajtai, $\sum_1^1$-formulae on finite structures, *Annals of Pure and Applied Logic* **24**, 1983, pp. 1–48.

[All89] E. Allender, A note on the power of threshold circuits, *Proc. 30th IEEE Symp. on Foundations of Computer Science*, 1989, pp. 580–584.

[Bop] R. Boppana, Private communication.

[FSS84] M. Furst, J. Saxe, and M. Sipser, Parity, circuits and the polynomial time hierarchy, *Mathematical Systems Theory* **17**, 1984, pp. 13–27.

[Haj88] P. Hajnal, *An $\Omega(N^{\frac{4}{3}})$ lower bound on the randomized complexity of graph properties*, Technical Report 88-19, University of Chicago, August 1988.

[HMP*87] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and Gy. Turán, Threshold circuits of bounded depth, *Proc. 28th IEEE Symp. on Foundations of Computer Science*, 1987, pp. 99–110.

[Hop82] J.J Hopfield, Neural network and physical systems with emergent collective computational abilities, *National Acad. Sci. USA* **79**, 1982, pp. 2554–2558.

[Kin88] V. King, Lower bounds on the complexity of graph properties, *Proc. 20th ACM Symp. on Theory of Computing*, 1988, pp. 468–476.

[LMN89] N. Linial, Y. Mansour, and N. Nisan, Constant depth circuits, fourier transform, and learnability, *Proc. 30th IEEE Symp. on Foundations of Computer Science*, 1989, pp. 574–579.

[Nis89] N. Nisan, CREW PRAMs and decision trees, *Proc. 21st ACM Symp. on Theory of Computing*, 1989, pp. 327–335.

[Raz87] A. A. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, *Mathematical Notes of the Academy of Sciences of the USSR* **41**, 1987, pp. 333–338.

[Rei87] J. Reif, On threshold circuits and polynomial computations, *2-nd structure in complexity theory conf.*, 1987, pp. 118–125.

[RMP86] D.E. Rumelhardt, J.L. McClelland, and the PDP research group, *Parallel Distributed Processing : Exploration in the Microstructure of Cognition*, Volume 1, MIT Press, 1986.

[RV78] R. Rivest and S. Viullemin, On recognizing graph properties from adjacency matrices, *Theoretical Computer Science* **3**, 1978, pp. 371–384.

[SFB] C. Sturtivant, G. Frandsen, and J. Boyar, Is finite field arithmetic a restricted model of computation? Manuscript, 1989.

[Smo87] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, *Proc. 19th ACM Symp. on Theory of Computing*, 1987, pp. 77–82.

[Sni85] M. Snir, Lower bounds for probabilistic linear decision trees, *Theoretical Computer Science* **38**, 1985, pp. 69–82.

[SW86] M. Saks and A. Wigderson, Probabilistic Boolean decision trees and the complexity of evaluating game trees, *Proc. 27th IEEE Symp. on Foundations of Computer Science*, 1986, pp. 29–38.

[Tod89] S. Toda, On the computational power of PP and $\oplus$P, *Proc. 30th IEEE Symp. on Foundations of Computer Science*, 1989, pp. 514–519.

[Yao87] A. C. Yao, Lower bounds to randomized algorithm for graph properties, *Proc. 28th IEEE Symp. on Foundations of Computer Science*, 1987, pp. 393–400.

[Yao89] A. C. Yao, Circuits and Local Computation, *Proc. 21st ACM Symp. on Theory of Computing*, 1989, pp. 186–196.