An improved lower bound for the randomized decision tree complexity of recursive majority[⋆] [⋆⋆]

Nikos Leonardos

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
nikos.leonardos@gmail.com

Abstract. We prove that the randomized decision tree complexity of the recursive majority-of-three is $\Omega(2.55^d)$, where d is the depth of the recursion. The proof is by a bottom up induction, which is same in spirit as the one in the proof of Saks and Wigderson in their 1986 paper on the complexity of evaluating game trees. Previous work includes an $\Omega\big((7/3)^d\big)$ lower bound, published in 2003 by Jayram, Kumar, and Sivakumar. Their proof used a top down induction and tools from information theory. In 2011, Magniez, Nayak, Santha, and Xiao, improved the lower bound to $\Omega\big((5/2)^d\big)$ and the upper bound to $O(2.64946^d)$.

Keywords: Boolean functions, randomized computation, decision tree complexity, query complexity, lower bounds, generalized costs

1 Introduction

In this paper we will be working with the decision tree model. We prove a lower bound on the randomized decision tree complexity of the recursive majority-of-three function. Formally, $\mathrm{maj}_1(x_1, x_2, x_3)$ is 1 if and only if at least two of x_1, x_2, x_3 are 1. Letting $y_i = (x_{(i-1)3^d+1}, \ldots, x_{i3^d})$, for $i = 1, 2, 3$, define for $d > 0$,

$$\mathrm{maj}_{d+1}(x_1, \ldots, x_{3^{d+1}}) = \mathrm{maj}(\mathrm{maj}_{d-1}(y_1), \mathrm{maj}_{d-1}(y_2), \mathrm{maj}_{d-1}(y_3)).$$

We write maj for maj_1. The function can be represented by a uniform ternary tree. In particular, let U_d be a tree of depth d, such that every internal node has three children and all leaves are on the same level. The function computed by interpreting U_d as a circuit with internal nodes labeled by maj-gates is maj_d.

This function seems to have been given by Ravi Boppana (see Example 1.2 in [7]) as an example of a function that has deterministic complexity 3^d, while its randomized complexity is asymptotically smaller. Other functions with this property are known. A notable example is the function nand_d, first analyzed by Snir [10]. This is the function represented by a uniform binary tree of depth d, with the internal nodes labeled by nand-gates. A simple randomized framework that can be used to compute both maj_d and nand_d is the following. Start at the root; as long as the output is not known, choose a child at random and evaluate it recursively. Algorithms of this type are called in [7] *directional.* For maj_d the

⋆ This research was partly supported by ERC project CODAMODA.
⋆⋆ Available online at ECCC: http://eccc.hpi-web.de/report/2012/099/.

directional algorithm computes the output in $(8/3)^d$ queries. It was noted in [7] that better algorithms exist for maj_d. Interestingly, Saks and Wigderson show that the directional algorithm is optimal for the nand_d function, and show that its zero-error randomized decision tree complexity is $\Theta\big(\big(\frac{1+\sqrt{33}}{4}\big)^d\big)$. Their proof uses a bottom up induction and generalized costs. Their method of generalized costs allows them to charge for a query according to the value of the variable. Furthermore, they conjecture that the maximum gap between deterministic and randomized complexity is achieved for this function.

Inspired by their technique we prove an $\Omega(2.55^d)$ lower bound on maj_d that also holds for algorithms with bounded-error. (The bound of [7] for nand_d was extended to bounded-error algorithms by Santha in [8].) In contrast to the exact asymptotic bounds we have for nand_d, there had been no progress on the randomized decision tree complexity of maj_d for several years. However, recent papers have narrowed the gap between the upper and lower bounds for recursive majority. An $\Omega\big((7/3)^d\big)$ lower bound was showed in [4]. Jayram, Kumar, and Sivakumar, proved their bound using tools from information theory and a top down induction. Furthermore, they presented a non-directional algorithm that improves the $O\big((8/3)^d\big)$ upper bound. Magniez, Nayak, Santha, and Xiao [6], significantly improved the lower bound to $\Omega\big((5/2)^d\big)$ and the upper bound to $O(2.64946^d)$. (Both of these lower bounds hold for the case that the randomized decision tree is allowed to err.)

Our proof of the lower bound is simpler than the aforementioned ones; it doesn't require a background in information theory and it only uses induction. Note that, Landau, Nachmias, Peres, and Vanniasegaram [5], showed how to remove the information theoretic notions from the proof in [4], keeping its underlying structure the same. Our proof can be even more simplified, if one requires the known $\Omega(2.5^d)$ lower bound. A simpler proof of this bound seems to have been already known to Jonah Serman [9] in 2007.

We note that both maj_d and nand_d, belong to the class of *read-once functions.* These are functions that can be computed by read-once Boolean formulae, that is, formulae such that each input variable appears exactly once. Heiman, Newman, and Wigderson [2] showed that read-once formulae with threshold gates have zero-error randomized complexity $\Omega(n/2^d)$ (here n is the number of variables and d the depth of a canonical tree-representation of the read-once function). Heiman and Wigderson [3] managed to show that for every read-once function f we have $R(f) \in \Omega(D(f)^{0.51})$, where $R(f)$ and $D(f)$ are the randomized and deterministic complexity of f respectively. Note that the conjecture of Saks and Wigderson states that for every function f we have $R(f) \in \Omega(D(f)^{0.753\ldots})$.

2 Definitions and notation

In this section we introduce basic concepts related to decision tree complexity. The reader can find a more complete exposition in the survey of Buhrman and de Wolf [1].

2.1 Definitions pertaining to decision trees

A *deterministic Boolean decision tree* Q over a set of variables $Z = \{z_i \mid i \in [n]\}$, where $[n] = \{1, 2, \ldots, n\}$, is a rooted and ordered binary tree. Each internal node is labeled by a variable $z_i \in Z$ and each leaf with a value from $\{0,1\}$. An *assignment* to Z (or an *input* to Q) is a member of $\{0,1\}^n$. The output $Q(\sigma)$ of Q on an input σ is defined recursively as follows. Start at the root and let its label be z_i. If $\sigma_i = 0$, we continue with the left child of the root; if $\sigma_i = 1$, we continue with the right child of the root. We continue recursively until we reach a leaf. We define $Q(\sigma)$ to be the label of that leaf. When we reach an internal node, we say that Q *queries* or *reads* the corresponding variable. We say that Q *computes* a Boolean function $f : \{0,1\}^n \to \{0,1\}$, if for all $\sigma \in \{0,1\}^n$, $Q(\sigma) = f(\sigma)$. The *cost of Q on input σ*, $\mathrm{cost}(Q;\sigma)$, is the number of variables queried when the input is σ. The *cost of Q*, $\mathrm{cost}(Q)$, is its *depth*, the maximum distance of a leaf from the root. The *deterministic complexity*, $D(f)$, of a Boolean function f is the minimum cost over all Boolean decision trees that compute f.

A *randomized Boolean decision tree* Q_{R} is a distribution p over deterministic decision trees. On input σ, a deterministic decision tree is chosen according to p and evaluated. The *cost of Q_{R} on input σ* is $\mathrm{cost}(Q_{\mathrm{R}};\sigma) = \sum_Q p(Q)\,\mathrm{cost}(Q;\sigma)$. The *cost* of Q_{R} is $\max_\sigma \mathrm{cost}(Q_{\mathrm{R}};\sigma)$. A randomized decision tree Q_{R} *computes* a Boolean function f, if $p(Q) > 0$ only when Q computes f. A randomized decision tree Q_{R} *computes* a Boolean function f *with error δ*, if, for all inputs σ, $Q_{\mathrm{R}}(\sigma) = f(\sigma)$ with probability at least $1-\delta$. The *randomized complexity*, $R(f)$, of a Boolean function f is the minimum cost of any randomized Boolean decision tree that computes f. The *δ-error randomized complexity*, $R_\delta(f)$, of a Boolean function f, is the minimum cost of any randomized Boolean decision tree that computes f with error δ.

We are going to take a distributional view on randomized algorithms. Let μ be a distribution over $\{0,1\}^n$ and Q_{R} a randomized decision tree. The *expected cost of Q_{R} under μ* is

$$\mathrm{cost}_\mu(Q_{\mathrm{R}}) = \textstyle\sum_\sigma \mu(\sigma)\,\mathrm{cost}(Q_{\mathrm{R}};\sigma).$$

The *δ-error expected complexity under μ*, $R^\mu_\delta(f)$, of a Boolean function f, is the minimum expected cost under μ of any randomized Boolean decision tree that computes f with error δ. Clearly, $R_\delta(f) \geq R^\mu_\delta(f)$, for any μ, and thus we can prove lower bounds on randomized complexity by providing lower bounds for the expected cost under any distribution.

2.2 Introducing cost-functions

We are going to utilize the method of generalized costs of Saks and Wigderson [7]. To that end, we define a *cost-function* relative to a variable set Z, to be a function $\phi : \{0,1\}^n \times Z \to \mathbb{R}$. We extend the previous cost-related definitions as follows. The cost of a decision tree Q under cost-function ϕ on input σ is

$$\mathrm{cost}(Q;\phi;\sigma) = \textstyle\sum_{z\in S} \phi(\sigma;z),$$

where $S = \{z \mid z \text{ is queried by } Q \text{ on input } \sigma\}$. The cost of a randomized decision tree Q_{R} on input σ under cost-function ϕ is

$$\mathrm{cost}(Q_{\mathrm{R}};\phi;\sigma) = \textstyle\sum_Q p(Q)\,\mathrm{cost}(Q;\phi;\sigma),$$

where p is the corresponding distribution over deterministic decision trees. Finally, the expected cost of a randomized decision tree Q_{R} under cost-function ϕ and distribution μ is

$$\mathrm{cost}_{\mu}(Q_{\mathrm{R}};\phi) = \textstyle\sum_{\sigma} \mu(\sigma)\,\mathrm{cost}(Q_{\mathrm{R}};\phi;\sigma).$$

Fact 1. *Let ϕ and ψ be two cost-functions relative to Z. For any decision tree Q over Z, any assignment σ to Z, and any $a, b \in \mathbb{R}$, we have*

$$a\,\mathrm{cost}(Q;\phi;\sigma) + b\,\mathrm{cost}(Q;\psi;\sigma) = \mathrm{cost}(Q;a\phi + b\psi;\sigma).$$

For $\phi, \psi : \{0,1\}^n \times Z \to \mathbb{R}$, we write $\phi \geq \psi$, if for all $(\sigma, z) \in \{0,1\}^n \times Z$, $\phi(\sigma, z) \geq \psi(\sigma, z)$.

Fact 2. *Let ϕ and ψ be two cost-functions relative to Z. For any decision tree Q over Z and any assignment σ to Z, if $\phi \geq \psi$, then* $\mathrm{cost}(Q;\phi;\sigma) \geq \mathrm{cost}(Q;\psi;\sigma)$.

2.3 Definitions pertaining to trees

For a rooted tree T, the *depth* of a leaf is the number of edges on the path to the root. The *depth* of the tree is the maximum depth of a leaf. We denote by L_T the set of its leaves and by V_T the set of its internal nodes. Define the set of *leaf-parents* of T, P_T, as the set of all nodes in V_T all of whose children are leaves. For $S \subseteq P_T$ let $L_T(S)$ be the set of the leaves of the nodes in S. We call a tree *uniform* if all the leaves are on the same level. A tree such that every node has exactly three children is called *ternary*. For a positive integer d, let U_d denote the uniform ternary tree of depth d.

In the following, let T denote a ternary tree with n leaves. We define a distribution μ_T over $\{0,1\}^n$ that is placing positive weight on inputs that we consider difficult; we call these inputs reluctant, in accordance with [7]. Let

$$M_0 = \{(0,0,1),(0,1,0),(1,0,0)\} \text{ and } M_1 = \{(0,1,1),(1,0,1),(1,1,0)\}.$$

In the following definition we view T as a circuit with every internal node labeled by a maj-gate. We denote the corresponding function by F_T.

Definition 3. *Call an input to a ternary tree* reluctant*, if it is such that the inputs to every gate belong to $M_0 \cup M_1$. Let μ_T, the* reluctant distribution for T*, be the uniform distribution over all reluctant inputs. We write $\mu_d \equiv \mu_{U_d}$, and $\mu_T(0)$ ($\mu_T(1)$) for μ_T conditioned on the output of F_T being* 0 (1).

Suppose the inputs to a gate, under an assignment σ, belong to M_0 (M_1). We call an input to this gate a *minority under* σ if it has the value 1 (0) and a *majority* otherwise.

3 Proof outline and preliminaries

Our goal is to prove a lower bound on the expected cost of any randomized decision tree Q_{R} that computes maj_d with bounded error δ. We now discuss the outline of our proof. We start with the tree $T \equiv U_d$ that represents maj_d, the natural cost-function ψ that charges 1 for any query, and the reluctant distribution $\mu \equiv \mu_T$. We define a process that shrinks tree T to a smaller tree T' and a corresponding randomized decision tree Q'_{R} that computes $F_{T'}$ with bounded error δ. The crucial part is to show that for a "more expensive" cost-function ψ', $\mathrm{cost}_\mu(Q_{\mathrm{R}}; \psi) \geq \mathrm{cost}_{\mu'}(Q'_{\mathrm{R}}; \psi')$, where $\mu' \equiv \mu_{T'}$. Our goal is to apply the shrinking process repeatedly to the leaves of U_d, until we obtain a recurrence of the form

$$R_\delta^{\mu_d}(\mathrm{maj}_d) \geq \lambda \cdot R_\delta^{\mu_{d-1}}(\mathrm{maj}_{d-1}),$$

for some constant λ. The quality of our lower bound (i.e. the constant λ, since the recurrence will lead to $R_\delta(\mathrm{maj}_d) \in \Omega(\lambda^d)$) will depend on how much more expensive ψ' is than ψ.

The main ingredient in this framework is the shrinking process. A natural choice would be to shrink T by removing three leaves u, v, w so that their parent s would become a leaf in T'. Then, if we had a good algorithm Q for F_T we could design an algorithm Q' for $F_{T'}$ as follows. On input σs, Q' would simulate Q on one of the inputs $\sigma 01s$, $\sigma 10s$, $\sigma 0s1$, $\sigma 1s0$, $\sigma s01$, $\sigma s10$, with equal probability. We will show in the next section that such a shrinking process can give an alternate—and simpler—proof of the $\Omega(2.5^d)$ lower bound of Magniez, Nayak, Santha, and Xiao [6].

To improve their bound we are going to shrink nine leaves to three at a time instead of three to one. This is made precise by the following definition.

Definition 4 **(**$\mathrm{shrink}(T;s)$**).** *For a ternary tree T, let s be the parent of $u, v, w \in P_T$. Define $\mathrm{shrink}(T;s)$ as the tree with the children of u, v, w removed.*

After shrinking our initial tree T to $T' \equiv \mathrm{shrink}(T;s)$ (notice that $u, v, w \in L_{T'}$), we need to define a randomized decision tree Q'_{R} that will compute $F_{T'}$ with error at most δ. We do so by defining for each deterministic tree Q that Q_{R} may choose, a randomized tree Q'.

Definition 5. *Let Q be any deterministic decision tree for F_T. We define a randomized decision tree Q' for $F_{T'}$, where $T' \equiv \mathrm{shrink}(T;s)$. The algorithm Q' on input σuvw chooses σ_u, σ_v, σ_w independently and uniformly at random from $\{x01, x10, 0x1, 1x0, 01x, 10x\}$, where x is u, v, w respectively. Then, Q' simulates Q on input $\hat{\sigma} = \sigma\sigma_u\sigma_v\sigma_w$. This induces a randomized algorithm Q'_{R} for $F_{T'}$.*

Fact 6. *If Q_{R} is a δ-error randomized decision tree for F_T, then Q'_{R} is a δ-error randomized decision tree for $F_{T'}$.*

It will be useful to express $\mathrm{cost}_{\mu'}(Q'; \psi')$, for some cost-function ψ', in terms of Q. We have the following proposition.

Proposition 7. *For a ternary tree T and s the parent of $u, v, w \in P_T$, let $T' \equiv \mathrm{shrink}(T;s)$. Let ψ' be a cost-function on T', such that $\psi'(\sigma; z) = \lambda$ for all $\sigma \in \{0,1\}^{|L_{T'}|}$ and $z \in \{u, v, w\}$. Then*

$$\mathrm{cost}_{\mu'}(Q'; \psi') = \mathrm{cost}_\mu(Q; \psi^*),$$

where (μ, μ') *is any of* $(\mu_T, \mu_{T'})$, $(\mu_T(0), \mu_{T'}(0))$, $(\mu_T(1), \mu_{T'}(1))$, *and*

$$\psi^*(\sigma; z) = \begin{cases} \psi'(\sigma; z), & \textit{if } z \in L_T \setminus L_T(u,v,w); \\ 0.5 \cdot \lambda, & \textit{if } z \in L_T(u,v,w) \textit{ and } z \textit{ is a majority under } \sigma; \\ 0, & \textit{if } z \in L_T(u,v,w) \textit{ and } z \textit{ is a minority under } \sigma. \end{cases} \tag{1}$$

Proof. Observe that by the definition of Q', $\Pr[\hat{\sigma} = \sigma] = \mu(\sigma)$. Furthermore, each σ is encountered 2^3 times over the random choices of Q'. For $(i,j,k) \in [3]^3$ define cost-functions for T as follows.

$$\psi_{(i,j,k)}(\sigma; z) = \begin{cases} 0, & \text{if } z \in L_T(u,v,w) \setminus \{u_i, v_j, w_k\}; \\ \lambda, & \text{if } z \in \{u_i, v_j, w_k\}; \\ \psi'(\sigma; z), & \text{otherwise.} \end{cases}$$

The indices i, j, k are going to play the role of x in Definition 5. We have

$$\text{cost}_{\mu'}(Q'; \psi') = \textstyle\sum_\sigma \mu(\sigma) \frac{1}{2^3} \sum_{(i,j,k) \in I_\sigma} \text{cost}(Q; \psi_{(i,j,k)}; \sigma),$$

where $I_\sigma = \{(i,j,k) \in [3]^3 \mid u_i, v_j, w_k \text{ are majorities under } \sigma\}$. The proposition follows since for any σ in the support of μ and any z, we have $\psi^*(\sigma, z) = \sum_{(i,j,k) \in I_\sigma} \frac{1}{2^3} \psi_{(i,j,k)}(\sigma, z)$. □

4 The $\Omega(2.5^d)$ lower bound and a toy problem

We sketch a proof of the $\Omega(2.5^d)$ lower bound of Magniez, Nayak, Santha, and Xiao [6], by applying the proof outline discussed in the previous section coupled with a simple shrinking process that shrinks three leaves to one. In addition, we define and analyze a toy problem that will play a crucial role in obtaining the improved $\Omega(2.55^d)$ bound.

For both of these tasks it is useful to define a cost-function ϕ_η, where $\eta \in \mathbb{R}$, as follows.

$$\phi_\eta(\sigma; z) = \begin{cases} 1, & \text{if } z \text{ is a minority under } \sigma; \\ \eta, & \text{otherwise.} \end{cases} \tag{2}$$

4.1 Proof of the $\Omega(2.5^d)$ lower bound

Let T be any ternary tree and x, y, z three of its leaves with a common parent u. Let T' be the ternary tree obtained by removing x, y, z, and thus transforming u to a leaf. Let ψ be a cost-function for T with $\psi(\sigma, x) = \psi(\sigma, y) = \psi(\sigma, z) = \lambda$ for any σ. Let ψ' be a cost-function for T' with $\psi'(\sigma, u) = 2.5 \cdot \lambda$ and $\psi'(\sigma, v) = \psi(\sigma, v)$ for any other leaf v. For any algorithm Q for F_T consider the algorithm Q' for $F_{T'}$ that on input σu outputs one of $Q(\sigma 01u)$, $Q(\sigma 10u)$, $Q(\sigma 0u1)$, $Q(\sigma 1u0)$, $Q(\sigma u01)$, $Q(\sigma u10)$ with equal probability. If Q is a δ-error algorithm for F_T, then Q' is a δ-error algorithm for $F_{T'}$. We claim that, with $\mu = \mu_T$ and $\mu' = \mu_{T'}$,

$$\text{cost}_\mu(Q; \psi) \geq \text{cost}_{\mu'}(Q'; \psi').$$

Accepting this claim, we start with $T \equiv U_d$ and apply it repeatedly by shrinking each time three leaves at depth d that are siblings. We end up with U_{d-1} and a cost function that charges 2.5 for each query. We have shown $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq 2.5 \cdot R_\delta^{\mu_{d-1}}(\mathrm{maj}_{d-1})$. Repeating this d times we obtain $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq 2.5^d \cdot R_\delta^{\mu_0}(\mathrm{maj}_0)$. Finally, it is not hard to show that you have to read a bit with probability at least $1 - 2\delta$ to be able to guess it with error at most δ, thus $R_\delta^{\mu_0}(\mathrm{maj}_0) \geq (1 - 2\delta)$. Putting these together, $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq (1 - 2\delta) \cdot 2.5^d$.

To prove the claim, we observe that as in Proposition 7, $\mathrm{cost}_{\mu'}(Q'; \psi') = \mathrm{cost}_\mu(Q; \psi'')$, where

$$\psi''(\sigma; z) = \begin{cases} \psi'(\sigma; z), & \text{if } z \in L_T \setminus L_T(u); \\ 1.25 \cdot \lambda, & \text{if } z \in L_T(u) \text{ and } z \text{ is a majority under } \sigma; \\ 0, & \text{if } z \in L_T(u) \text{ and } z \text{ is a minority under } \sigma. \end{cases}$$

By Fact 1 it suffices to show that $\mathrm{cost}_\mu(Q; \psi - \psi'') \geq 0$. Note now that $\psi - \psi''$ is equal to $\lambda\phi_{-0.25}$ on x, y, z, and zero everywhere else. Thus, we can focus on how these three leaves are queried by Q and ignore all the other leaves. To do this, observe that if we fix values on the rest of the leaves, we obtain from Q a decision tree on three variables.

In the table below we list the deterministic decision trees[1] Q for three variables that are relevant to our problem. We label the input variables x, y, z, in the order they are queried. We write "and* z" to denote a conditional read. That is, z is queried only if the value of $\mathrm{maj}(x, y, z)$ cannot be determined from the values of x and y. Decision trees that read z even if $x = y$ are of no interest, neither for maj_d, nor for the toy problem we will consider in the next section.

In the last column we calculate $\sum_{\sigma \in M_0} \mathrm{cost}(Q; \phi_\eta; \sigma)$. Because of the symmetries involved we can look up the costs for $\sigma \in M_1$ as well. For example, the cost of the decision tree in row (2a) when $\sigma \in M_1$, is the same as the cost of the decision tree in row (2b) when $\sigma \in M_0$.

	Decision tree	Cost
(1)	If $x = 0$, stop; if $x = 1$, stop.	$1 + 2\eta$
(2a)	If $x = 0$, stop; if $x = 1$, read y.	$1 + 3\eta$
(2b)	If $x = 0$, read y; if $x = 1$, stop.	$2 + 3\eta$
(3a)	If $x = 0$, stop; if $x = 1$, read y and* z.	$1 + 4\eta$
(3b)	If $x = 0$, read y and* z; if $x = 1$, stop.	$2 + 4\eta$
(4)	If $x = 0$, read y; if $x = 1$, read y.	$2 + 4\eta$
(5a)	If $x = 0$, read y; if $x = 1$, read y and* z.	$2 + 5\eta$
(5b)	If $x = 0$, read y and* z; if $x = 1$, read y.	$2 + 5\eta$
(6)	If $x = 0$, read y and* z; if $x = 1$, read y and* z.	$2 + 6\eta$

What we are going to use from this table is that for $\eta \in [-0.5, 0]$ the decision tree of row (3a) has the minimum cost when $(x, y, z) \in M_0$, and the tree of row (3b) when $(x, y, z) \in M_1$. Their cost is $1 + 4\eta$. One can now verify that there is

[1] We abuse the term "decision tree" here, since we are actually listing algorithms that query bits but do not output anything.

no (deterministic) decision tree Q that can achieve $\mathrm{cost}_\mu(Q;\phi_{-0.25}) < 0$. This completes the proof of the claim and the sketch of the $\Omega(2.5^d)$ lower bound.

Remark. Note the role of the value of u in the above argument. In particular, if $u = 0$, then the best decision tree is the one on row (3a), whereas if $u = 1$, it is the one on row (3b). We can do better, if we only want a bound for maj_1.

Proposition 8. $R_\delta^{\mu_1}(\mathrm{maj}_1) \geq \frac{8}{3} \cdot R_\delta^{\mu_0}(\mathrm{maj}_0)$.

Proof. Let ψ_1 be the cost-function for maj_1 defined by $\psi_1(\sigma; z) = 1$ for all σ and z. Let ψ_0 be the cost-function for maj_0 defined by $\psi_0(0; u) = \psi_0(1; u) = 8/3$. Then, as in the proof of Proposition 7, we can show that $\mathrm{cost}_{\mu_1}(Q_1; \psi_1) - \mathrm{cost}_{\mu_0}(Q_0; \psi_0) = \mathrm{cost}_{\mu_1}(Q_1; \phi_{-1/3})$. Observe now—by examining the table—that for any deterministic algorithm Q, $\sum_{\sigma \in M_0 \cup M_1} \mathrm{cost}(Q; \phi_{-1/3}; \sigma) \geq 0$. The zero is achieved by the tree on row (6) of the table. Thus, $\mathrm{cost}_{\mu_1}(Q_1; \psi_1) \geq \mathrm{cost}_{\mu_0}(Q_0; \psi_0)$ and the result follows. □

4.2 The toy problem and a corollary

Recall that in order to improve the lower bound, we need a shrinking process that shrinks nine leaves to three. Since it would be rather tedious to analyze decision trees on nine variables, we introduce a toy problem that reduces our analysis on decision trees over $\{0,1\}^6$. We present first the toy problem and following its analysis a corollary that reveals its usefulness.

Let μ be the uniform distribution over $\{(u, v) \mid (u \in M_0 \wedge v \in M_1) \vee (u \in M_1 \wedge v \in M_0)\}$. We seek the minimum real η for which $\mathrm{cost}_\mu(Q; \phi_\eta) \geq 0$ for any decision tree Q. We show that we can have $\eta = -0.3$. Although it is not stated in the following lemma, it is easily observable from the proof that this value is best possible. Although the analysis of the toy problem is optimal, one could improve the constant 2.55 by analyzing directly the decision trees over $\{0,1\}^9$.

Lemma 9. *For any decision tree Q over $\{0,1\}^6$, $\mathrm{cost}_\mu(Q; \phi_{-0.3}) \geq 0$.*

Proof. For the proof we are going to do some case analysis, taking advantage of the symmetries involved. Denote the input by (x, y, z, u, v, w), and call (x, y, z) the left side and (u, v, w) the right side. Assume, without loss of generality (due to the symmetry of μ and the fact that we are calculating expected cost), that the variables on the left side are queried in the order x, y, z and on the right side in the order u, v, w. Assume further, that x is the first variable queried by Q, and let Q_0 (Q_1) be the decision tree if $x = 0$ ($x = 1$). We only analyze Q_0, as the analysis of Q_1 would be the same with the roles of 0 and 1 exchanged. Thus, we assume $x = 0$ and proceed with the analysis of Q_0.

In all of the following cases we calculate the cost scaled; in particular, we calculate $C \equiv \sum_{\sigma : x=0} \mathrm{cost}(Q; \phi; \sigma)$.

Case 1. Suppose that Q_0 is empty. Then $C = 3 + 6\eta > 0$.

Case 2. Suppose that Q_0 queries y. Then, either $x = y$ or $x \neq y$. In the first case, we may assume Q_0 does not query z, since such a query increases the cost by 1. In the second case, we may assume Q_0 queries z, since such a query decreases the cost by $-\eta$. Therefore, the optimal Q_0 first "finishes" with the left side and then proceeds to the right side, knowing whether $(u, v, w) \in M_0$ or $(u, v, w) \in M_1$. In the first case, the optimal Q_0 continues with the right side as in row (3a) of the table; in the second case, as in row (3b). The cost is $C = (3 \cdot 2\eta + 1 + 4\eta) + 2 \cdot (3 \cdot (1 + 2\eta) + 1 + 4\eta)$, which is 0 for $\eta = -0.3$.

Case 3. Suppose that Q_0 queries u.

(i) Suppose $x = u$. If Q_0 does not query anything else, then this case contributes to the cost $4\cdot(1+\eta)$. Otherwise let us assume (without loss of generality) that it reads y. Then, as in Case 2, we may assume that Q_0 "finishes" the left side before doing anything else. There are four inputs such that $x = u = 0$. For two of the inputs the left side belongs to M_0 and for the other two to M_1. In the first case, the optimal Q_0 reads v and w (they are both majorities). In the second case, it does not read any of v, w (it costs an additional $1 + 2\eta > 0$ if it reads them). In total the cost of this case is then $(1+4\eta)+(2+4\eta)+2\cdot(1+3\eta) = 5+14\eta$.

(ii) Suppose $x \neq u$. If Q_0 does not query anything else, then this case contributes to the cost $2 + 8\eta$. Otherwise let us assume (without loss of generality) that it reads y. With similar considerations as in case 3(i), we find that the total cost of this case is then $(2 + 4\eta) + 2 \cdot 3\eta + 2 \cdot (1 + 3\eta) = 4 + 16\eta$.

Summing up for case 3, we find that the best Q_0 can do is $C = 9 + 30\eta = 0$. The case analysis is complete. □

We prove a corollary of this lemma that connects the toy problem to our real goal, which is the analysis of the process of shrinking nine leaves to three.

Lemma 10. *Let $T \equiv U_2$ with root s and $T' \equiv \mathrm{shrink}(T; s)$. Let ψ and ψ' be cost-functions such that $\psi(\sigma; z) = \lambda \geq 0$ for all $\sigma \in \{0,1\}^9$ and all variables $z \in L_T$, and $\psi'(\sigma; z) = 2.55 \cdot \lambda$ for all $\sigma \in \{0,1\}^3$ and all variables $z \in L_{T'}$. Then, for any deterministic decision tree Q over $\{0,1\}^9$,*

$$\mathrm{cost}_\mu(Q; \psi) \geq \mathrm{cost}_{\mu'}(Q'; \psi'),$$

where (μ, μ') is any of $(\mu_T, \mu_{T'})$, $(\mu_T(0), \mu_{T'}(0))$, $(\mu_T(1), \mu_{T'}(1))$.

Proof. Recall the definition of ψ^* from page 6. We have

$$\begin{aligned}
&\mathrm{cost}_\mu(Q; \psi) - \mathrm{cost}_{\mu'}(Q'; \psi') \\
&\quad = \mathrm{cost}_\mu(Q; \psi) - \mathrm{cost}_\mu(Q; \psi^*) && \text{by Proposition 7} \\
&\quad = \mathrm{cost}_\mu(Q; \psi - \psi^*) && \text{by Fact 1} \\
&\quad = \textstyle\sum_{\sigma:\mathrm{maj}_2(\sigma)=0} \mu(\sigma)\,\mathrm{cost}(Q; \psi - \psi^*; \sigma) \\
&\qquad\qquad + \textstyle\sum_{\sigma:\mathrm{maj}_2(\sigma)=1} \mu(\sigma)\,\mathrm{cost}(Q; \psi - \psi^*; \sigma).
\end{aligned}$$

According to whether we are interested in μ_T, $\mu_T(0)$, or $\mu_T(1)$, one of the sums might be empty. Without loss of generality, we assume the first sum is nonempty and show it is nonnegative. The other sum can be treated similarly. To that end, we define an intermediate cost-function ξ. In the following definition, σ is an assignment, z a variable, and u is the value of the parent of z under σ.

$$\xi(\sigma; z) = \begin{cases} \lambda, & \text{if } z \text{ is a minority under } \sigma; \\ -0.275 \cdot \lambda, & \text{if } z \text{ is a majority under } \sigma \text{ and } u = 0; \\ -0.3 \cdot \lambda, & \text{if } z \text{ is a majority under } \sigma \text{ and } u = 1. \end{cases}$$

Observe that $\psi - \psi^* \geq \xi$ (they agree on minorities and $\psi - \psi^*$ is $\lambda - 0.5 \cdot 2.55 \cdot \lambda = -0.275 \cdot \lambda$ on all majorities) and thus it suffices to show that

$$\textstyle\sum_{\sigma:\mathrm{maj}_2(\sigma)=0} \mu(\sigma) \operatorname{cost}(Q; \xi; \sigma) \geq 0. \tag{3}$$

We are going to decompose the above sum into terms that correspond either to the toy problem and Lemma 9 applies or correspond to queries over 3 variables and the table of Section 4.1 can be used.

To that end, we decompose ξ into several cost-functions. Let u, v, and w be the children of s. Define a cost-function ξ_u by

$$\xi_u(\sigma; z) = \begin{cases} 0, & \text{if } z \in L_T(u); \\ -0.3 \cdot \lambda, & \text{if } z \text{ is a majority under } \sigma \text{ and } z \in L_T(v, w); \\ \lambda, & \text{if } z \text{ is a minority under } \sigma \text{ and } z \in L_T(v, w). \end{cases}$$

Similarly define ξ_v and ξ_w. For $\alpha \in M_0$ define

$$C_u(\alpha) \equiv \textstyle\sum_{\beta \in M_0} \sum_{\gamma \in M_1} \mu(\alpha\beta\gamma) \operatorname{cost}(Q; \xi_u; \alpha\beta\gamma) + \mu(\alpha\gamma\beta) \operatorname{cost}(Q; \xi_u; \alpha\gamma\beta).$$

Similarly define C_v and C_w (assigning α to v and w respectively). These terms—as shown later—correspond to the toy problem. Define a cost-function ξ'_u by

$$\xi'_u(\sigma; z) = \begin{cases} 0, & \text{if } z \in L_T(v, w); \\ -0.25 \cdot \lambda, & \text{if } z \text{ is a majority under } \sigma \text{ and } z \in L_T(u); \\ \lambda, & \text{if } z \text{ is a minority under } \sigma \text{ and } z \in L_T(u). \end{cases}$$

Similarly define ξ'_v and ξ'_w. For $(\alpha, \beta) \in M_0 \times M_1$ define

$$C'_u(\alpha, \beta) \equiv \textstyle\sum_{\gamma \in M_0} \mu(\gamma\alpha\beta) \operatorname{cost}(Q; \xi'_u; \gamma\alpha\beta) + \mu(\gamma\beta\alpha) \operatorname{cost}(Q; \xi'_u; \gamma\beta\alpha).$$

Similarly define C'_v and C'_w. These terms will be analyzed using the table.

We now argue that we have the following decomposition

$$\begin{aligned} \textstyle\sum_{\sigma:\mathrm{maj}_2(\sigma)=0} \mu(\sigma) \operatorname{cost}(Q; \xi; \sigma) = \frac{1}{2} \Big[& \textstyle\sum_{\alpha \in M_0} \Big(C_u(\alpha) + C_v(\alpha) + C_w(\alpha) \Big) \\ & + \textstyle\sum_{\alpha \in M_0} \sum_{\beta \in M_1} \Big(C'_u(\alpha, \beta) + C'_v(\alpha, \beta) + C'_w(\alpha, \beta) \Big) \Big]. \end{aligned} \tag{4}$$

To prove this, we fix a $\sigma = xyz$ on the left-hand side and see if each bit—assuming it is queried by Q—is charged the same in both sides of the equation. Without loss of generality, let us assume σ is such that $\mathrm{maj}(x) = \mathrm{maj}(y) = 0$

and $\text{maj}(z) = 1$. A minority (under σ) is charged λ on the left side. A minority below u is charged λ once in $C_v(y)$ and once in $C'_u(y, z)$ on the right, for a total of $0.5 \cdot (\lambda + \lambda)$. Similarly for a minority below v. A minority below w will be charged λ in $C_u(x)$ and $C_v(y)$, which corresponds to the amount charged on the left side. A majority below u will be charged $-0.3 \cdot \lambda$ in $C_v(y)$ and $-0.25 \cdot \lambda$ in $C'_u(y, z)$, for a total of $0.5 \cdot (-0.3 - 0.25) \cdot \lambda$; this is how much is charged in the left side as well. Similarly for a majority below v. Finally, a majority below w is charged $-0.3 \cdot \lambda$ in $C_u(x)$ and $C_v(y)$, equal to the amount charged on the left side.

We now finish the proof by showing that the right-hand side of (4) is non-negative. We argue that, for any $\alpha \in M_0$, $C_u(\alpha) \geq 0$. Each fixed $\alpha \in M_0$ induces a decision tree Q_α over $\{0, 1\}^6$ such that $Q_\alpha(\beta\gamma) = Q(\alpha\beta\gamma)$. Observe that ξ_u agrees on $L_T(v, w)$ with $\lambda\phi_{-0.3}$ (where $\phi_{-0.3}$ is defined in Equation 2). Thus, $C_u(\gamma) = \lambda \operatorname{cost}_\mu(Q_\gamma; \phi_{-0.3})$ and Lemma 9 shows that $C_u(\alpha) \geq 0$. Similarly, for any $\alpha \in M_0$, $C_v(\gamma)$, $C_w(\gamma) \geq 0$. Along similar lines we can show that, for any $(\alpha, \beta) \in M_0 \times M_1$, $C'_u(\alpha, \beta)$, $C'_v(\alpha, \beta)$, $C'_w(\alpha, \beta) \geq 0$. (Lemma 9 is not needed in this case; inspection of the table in Section 4.1 suffices.) □

5 Proof of the lower bound

We prove a lemma that carries out the inductive proof sketched in Section 3.

Lemma 11 (Shrinking Lemma). *For a ternary tree T and s the parent of $u, v, w \in P_T$, let T' denote* $\text{shrink}(T; s)$*. Let ψ and ψ' be cost-functions on T and T' such that $\psi(\sigma; z) = \lambda$ for all $\sigma \in \{0, 1\}^{|L_T|}$ and $z \in L_T(u, v, w)$ and*

$$\psi'(\sigma; t) = \begin{cases} 2.55 \cdot \lambda, & \text{if } t \in \{u, v, w\}; \\ \psi(\sigma; t), & \text{otherwise.} \end{cases}$$

Then, for any randomized decision tree Q_R, $\mu \equiv \mu_T$ and $\mu' \equiv \mu_{T'}$,

$$\operatorname{cost}_\mu(Q_\text{R}; \psi) \geq \operatorname{cost}_{\mu'}(Q'_\text{R}; \psi').$$

Proof. Let n denote the number of leaves in T. Fix a partial assignment $\pi \in \{0, 1\}^{n-9}$ for the leaves in $L_T \setminus L_T(u, v, w)$ and let $\rho \in \{0, 1\}^9$. We write $\pi\rho$ for the assignment that equals ρ on the variables $L_T(u, v, w)$ and π everywhere else. For any deterministic tree Q we have (recalling Proposition 7 and Fact 1)

$$\begin{aligned} \Delta(Q) &\equiv \operatorname{cost}_\mu(Q; \psi) - \operatorname{cost}_{\mu'}(Q'; \psi') = \operatorname{cost}_\mu(Q; \psi) - \operatorname{cost}_\mu(Q; \psi^*) \\ &= \operatorname{cost}_\mu(Q; \psi - \psi^*) = \textstyle\sum_{\pi\rho} \mu(\pi\rho) \operatorname{cost}(Q; \psi - \psi^*; \pi\rho). \end{aligned}$$

Now, ψ and ψ^* are equal over $L_T \setminus L_T(u, v, w)$. Furthermore, having fixed π, we can define a deterministic tree Q_π over $\{0, 1\}^9$ so that on input $\rho \in \{0, 1\}^9$ we have $Q_\pi(\rho) = Q(\pi\rho)$. Thus $\Delta(Q) = \sum_{\pi\rho} \mu(\pi\rho) \operatorname{cost}(Q_\pi; \psi - \psi^*; \rho)$. Finally, recalling the definition of ψ^* (page 6), we see that $\psi - \psi^* \geq \lambda\phi_{-0.3}$ on $L_T(u, v, w)$ and by Fact 2, $\Delta(Q) \geq \sum_{\pi\rho} \mu(\pi\rho) \operatorname{cost}(Q_\pi; \lambda\phi_{-0.3}; \rho)$. Thus, we may apply Lemma 10, which implies that, for each fixed π, each summand is greater or equal to zero. It follows that, for any Q, $\Delta(Q) \geq 0$. We have $\operatorname{cost}_\mu(Q_\text{R}; \psi) - \operatorname{cost}_{\mu'}(Q'_\text{R}; \psi') = \sum_Q p(Q)\Delta(Q) \geq 0$. □

Theorem 12. $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq \frac{8}{3} \cdot (1-2\delta) \cdot 2.55^{d-1}$.

Proof. We start with $T \equiv U_d$ and apply the Shrinking Lemma repeatedly by shrinking each time nine leaves at depth d that have a common ancestor at depth $d-2$. We end up with U_{d-1} and a cost function that charges 2.55 for each query, obtaining $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq 2.55 \cdot R_\delta^{\mu_{d-1}}(\mathrm{maj}_{d-1})$. Repeating this $d-1$ times we get $R_\delta^{\mu_d}(\mathrm{maj}_d) \geq 2.55^{d-1} \cdot R_\delta^{\mu_1}(\mathrm{maj}_1)$. By Proposition 8, $R_\delta^{\mu_1}(\mathrm{maj}_1) \geq \frac{8}{3} \cdot R_\delta^{\mu_0}(\mathrm{maj}_0)$. A δ-error decision tree for maj_0 should guess a random bit with error at most δ; thus, $R_\delta^{\mu_0}(\mathrm{maj}_0) \geq 1-2\delta$. □

Acknowledgements

I thank Mike Saks for useful discussions and corrections in an earlier draft of this paper and Jeff Steif for pointing out a serious error in an earlier version. I also thank Carola Winzen and other reviewers for useful comments.

References

[1] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.
[2] Rafi Heiman, Ilan Newman, and Avi Wigderson. On read-once threshold formulae and their randomized decision tree complexity. *Theor. Comput. Sci.*, 107(1):63–76, 1993.
[3] Rafi Heiman and Avi Wigderson. Randomized vs. deterministic decision tree complexity for read-once boolean functions. *Computational Complexity*, 1:311–329, 1991.
[4] T. S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *STOC*, pages 673–682. ACM, 2003.
[5] Itamar Landau, Asaf Nachmias, Yuval Peres, and Sithparran Vanniasegaram. The lower bound for evaluating a recursive ternary majority function and an entropy-free proof. *Undergraduate Research Reports, Department of Statistics, University of California, Berkeley*, 2006.
[6] Frédéric Magniez, Ashwin Nayak, Miklos Santha, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 317–329. Springer, 2011.
[7] Michael E. Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *FOCS*, pages 29–38. IEEE Computer Society, 1986.
[8] Miklos Santha. On the monte carlo boolean decision tree complexity of read-once formulae. *Random Struct. Algorithms*, 6(1):75–88, 1995.
[9] Jonah Sherman. Communicated to the author by Ryan O'Donnell in 12 Sep. 2012. *Unpublished*, 2007.
[10] Marc Snir. Lower bounds on probabilistic linear decision trees. *Theor. Comput. Sci.*, 38:69–82, 1985.