arXiv:2301.00136v1 [cs.CC] 31 Dec 2022

单调查询决策树的能力

Prashanth Amireddy* Sai Jayasurya† Jayalal Sarma‡

2023年1月3日

摘要

本文首次研究自适应和非自适应单调决策树的计算能力——这类决策树的每个查询都是关于输入位的单调函数。在最一般的情形下，单调决策树高度（或规模）可以视为给定布尔函数非单调性的一种度量。我们还研究了对该模型的限制，即限制（从电路复杂度角度）每个节点可查询的单调函数。这自然引出了形如 DT(mon-$\mathcal{C}$) 的复杂度类，其中 $\mathcal{C}$ 是任意电路复杂度类，树的高度为 $\mathcal{O}(\log n)$，查询函数则可由类 $\mathcal{C}$ 中的单调电路计算。在上述背景下，我们证明了以下刻画与界。

- 对于任意布尔函数 f，我们证明，单调决策树的最小高度可以用其交替数精确刻画（在模型的自适应和非自适应版本中均如此；alt(f) 定义为该函数值在布尔格的任意一条链上发生变化的最大次数）。我们还用函数认证复杂度的一种自然推广刻画了非自适应决策树高度。同样，我们根据 alt(f) 确定了单调决策树的非确定性和随机化变体的复杂度。

- 我们证明，当 $\mathcal{C}$ 包含用于阈值函数的单调电路时，DT(mon-$\mathcal{C}$) $= \mathcal{C}$（例如，当 $\mathcal{C} =$ 为 TC^0 时）。对于 $\mathcal{C} = \mathrm{AC}_0$，我们能够证明，$\mathrm{AC}^0$ 中的任意函数都可以由一棵次线性高度的单调决策树计算，其中的查询由单调 AC^0 电路实现。

- 为了理解 AC^0 的对数高度情形，即 DT(mon-AC^0)，我们证明，对于 DT(mon-AC^0) 中任意一个作用于 n 比特的 f，以及任意正的常数 $\epsilon \leq 1$，都存在一个计算 f 的 AC^0 电路，其含有 $\mathcal{O}(n^\epsilon)$ 个否定门。

在证明主要结论的过程中，我们研究了决策列表模型的单调变体，并用 alt(f) 给出了相应的刻画；此外，我们还推导出一个结论：如果 $\mathcal{C}$ 具有适当的闭包性质，那么 DT(mon-$\mathcal{C}$) $=$ DL(mon-$\mathcal{C}$)（其中 DL(mon-$\mathcal{C}$) 的定义与 DT(mon-$\mathcal{C}$) 类似，只不过针对的是决策列表）。

*哈佛大学。此项工作完成于作者在印度理工学院马德拉斯分校本科就读期间。
†此项工作完成于作者在印度理工学院马德拉斯分校本科就读期间。
‡印度理工学院马德拉斯分校。

目录

1 引言 **2**

2 预备知识 **6**2.1 MDL 的一种标准形式.7

3 我们的工具：布尔函数的单调分解 **8**

3.1 特殊情形下交替分解的唯一性.103.2 关于单调分量的约束 TC^0 及更高层级.10

4 确定性自适应单调决策树 **13**

4.1 单调决策树与单调决策列表.134.2 自适应决策树高度的刻画.164.3 从否定受限电路构造自适应 MDL.16

5 确定性非自适应单调决策树 **17**

5.1 单调证书复杂度.18

6 非确定性单调决策树 **19**

6.1 两种等价定义.196.2 使用交替性刻画.21

7 随机化单调决策树 **21**

8 单调决 带查询限制的决策树**24**

8.1DT(单-$\mathcal{C}$) 上的高度约束.248.2 带查询限制的确定性MDT：DT(单-$\mathcal{C}$) 与 $\mathcal{C}$ .258.3 单调决策树与AC^0 .268.4 带查询限制的随机化MDT.29

9 讨论与开放问题**32**

A 附录 **34**

A.1 单调分解引理的证明（引理 3.1).34A.2 AC^0 有序输入的反相器构造——改编自 [18].35

1 引言

决策树模型是一种基础抽象，能够捕捉各种场景中的计算，从基于查询的决策过程到用于

布尔函数的学习算法。该模型将计算布尔函数 $f\colon \{0,1\}^n \to \{0,1\}$ 所需的算法步骤表示为一系列基于对输入位进行查询的分支操作，而分支取决于查询结果。很自然地，可以将这些分支视为一棵有根二叉树，树的叶节点标记为 0 或 1，表示计算到达该叶节点时函数的值。

所研究的最简单形式是直接查询输入 [9,15] 中的比特，因此，决策树的节点（叶节点除外）标记为其查询的输入变量。对于布尔函数 f，确定性决策树复杂度 $\mathsf{DT}(f)$ 是计算 f的所有决策树中的最小高度。这里的 高度始终指从根节点到叶节点的路径上内部节点的最大数量。决策树的规模定义为树中叶节点的数量，它是衡量 f复杂度的一个独立且有趣的指标。事实上，由于树是二进制的，其规模不可能超过 $\mathsf{DT}(f)$ 的指数级。算法领域也研究了决策树模型的推广形式，即随机决策树和量子决策树（见 [9]）。根据算法中下一次查询是否取决于先前查询的布尔结果，决策树可以是自适应的，也可以是非自适应的。在树的表示中，这对应于同一层的所有节点是否查询同一个变量。

（自适应）决策树高度 $\mathsf{DT}(f)$ 与布尔函数的许多基本复杂度度量有关。已知它与 f 在 $\mathbb{R}$ 上的次数、块敏感度、证书复杂度（参见综述 [9]）满足多项式关系；随着敏感度猜想 [14],最近得到证明，它甚至与布尔函数 f的敏感度满足多项式关系。非自适应决策树的能力则不那么强。

推广决策树模型的一种重要方式，是允许使用比单个位查询更强的查询。在这一方向上研究较多的一种模型是奇偶性决策树，其中每次查询都是输入位 [16]某个子集的奇偶性。树中的每个节点都与一个子集$S \subseteq [n]$ [1] 相关联，而在该节点对输入进行的查询是函数 $\oplus_{i\in S} x_i$，其中 x_i表示 x的 i^{th} 位。奇偶性决策树模型因其与对数秩猜想的一个特殊情形——异或对数秩猜想[24]——之间的联系而备受关注。该猜想尤其意味着，一般而言，函数的非自适应（$\mathsf{DT}^{\mathsf{na}}_{\oplus}(f)$）和自适应（$\mathsf{DT}_{\oplus}(f)$）奇偶性决策复杂度度量之间不存在多项式关系[2]。

标准决策树模型的其他经过充分研究的推广包括线性决策树 [10, 21, 23] （其中每个节点查询一个形如 $\sum_i \alpha_i x_i + \beta > 0$ 的线性函数）和代数决策树 [4, 5, 22] （其中每个节点查询输入变量的多项式求值的符号，且次数至多为 d ）。多项式规模的线性决策树可以计算 NP 完全的背包问题，而上述研究证明了显式语言的指数规模下界。Ben-Asher 和 Newman [3], 研究了允许合取

[1]我们将集合 $\{1, 2, \ldots, n\}$ 记为 $[n]$。[2]如果 $\mathsf{supp}(f) = \{S \subseteq [n] \mid \hat{f}(S) \neq 0\}$，$\mathsf{sps}(f) = |\mathsf{supp}(f)|$，且 $\mathsf{fdim}(f) = \dim(\mathsf{supp}(f))$，那么根据 [24],$\log \mathsf{sps}(f)/2 \leq \mathsf{DT}_{\oplus}(f) \leq \mathsf{fdim}(f) = \mathsf{DT}^{\mathsf{na}}_{\oplus}(f)$ [12,19]。异或对数秩猜想 [24] 指出，$\mathsf{DT}_{\oplus}(f) \leq \mathrm{poly}(\log \mathsf{sps}(f))$，并且存在 $\exists f$ 使得 $\mathsf{fdim}(f)$ 与 $\log(\mathsf{sps}(f))$ 相差呈指数级。

和析取变量作为内部节点查询的决策树，并证明了计算阈值函数所需的这类决策树的高度下界。

我们的结果：
我们开始研究一种新的决策树模型推广形式，该模型允许使用更一般的查询。我们的模型最一般的版本允许算法查询输入上的任意单调函数[3]。我们将函数 f的确定性单调决策树复杂度定义为 $\mathrm{DT}_m(f)$，即每个节点都进行单调查询且能够计算 f的决策树的最小高度。当决策树是非自适应的（即查询函数不依赖于先前查询的结果）时，我们用$\mathrm{DT}_m^{na}(f)$表示。

DT_m 和 DT_m^{na} 作为非单调性的度量：单调决策树复杂度度量也可以解释为函数 f非单调性的度量。我们的第一个结果是用一种广为研究的非单调性度量——交替数，对这一度量进行精确刻画。我们的首个主要结果揭示了自适应和非自适应情形下，单调决策树高度与函数交替数之间的如下联系。与奇偶性决策树情形中的猜想类似，两者相差呈指数级。

定理 1.1。　对于任意布尔函数 f，$\mathrm{DT}_m(f) = \lceil \log(\mathrm{alt}(f)+1) \rceil$，且 $\mathrm{DT}_m^{na}(f) = \mathrm{alt}(f)$.

在证明上述定理的过程中，我们还将其与一种经过深入研究的计算模型——决策列表（定义见第 2 节）的一种类似推广联系起来。若 $\mathrm{DL}_m(f)$ 表示计算布尔函数 f 的任意单调决策列表的最小长度，则我们证明 $\mathrm{DL}_m(f) = \mathrm{alt}(f)+1$。我们还给出了布尔函数证书复杂度的一种自然推广，记作 C_m（其非自适应版本记作 C_m^{na}），并证明对于每个函数 f，$\mathrm{C}_m^{na}(f) = \mathrm{DT}_m^{na}(f)$（命题 5.2）。

非确定性与随机化单调决策树：我们研究非确定性与随机化单调决策树（见第 6 节和 7 节），并考察定义的若干变体，给出等价关系和界。具体而言，我们证明非确定性单调决策树的高度具有常数上界（定理 6.1），并用确定性单调决策树复杂度刻画随机化版本的高度，从而也刻画交替数（定理 7.1）。

具有受限（单调）查询的决策树：虽然上述模型提供了一种衡量非单调性的方法，但上述决策树模型的主要缺点之一是，该计算模型无法简洁地表示。自然会想到，我们能否将查询函数限制为允许对函数进行简洁表示的电路复杂度类。一个直接的研究方向是了解将查询函数限制为电路复杂度类后，该模型的能力如何；相关内容见第 8 节。更正式地说，我们定义$\mathrm{DT}(\text{单调-}\mathcal{C})$ 为

[3]事实上，这个广　义化模型仍然是通用的，因为在普通决策树中，查询是单调函数　.

可由高度为 $\mathcal{O}(\log n)$ 的单调决策树计算的函数类，其中每个查询函数都具有属于 $\mathcal{C}$ 的单调电路，或者等价地，所有查询都属于单调-$\mathcal{C}$。

为了论证单调决策树高度的 $\mathcal{O}(\log n)$ 上界，我们证明：如果允许高度上界与 $\Theta(\log n)$ 在渐近意义上不同，那么该模型所能计算的函数类也会与 $\mathcal{C}$ [4] 不同。更准确地说，若 $\mathrm{DT}^{d(n)}$ 表示由高度至多为 $d(n)$ 的单调决策树所计算的函数类（因此 DT(mon-$\mathcal{C}$) $\equiv \mathrm{DT}^{\mathcal{O}(\log n)}$(mon-$\mathcal{C}$)），我们证明，对于任意 $g(n) = o(\log n)$ 和 $h(n) = \omega(\log n)$，都有 $\mathrm{DT}^{g(n)}$(mon-$\mathcal{C}$) $\subsetneq \mathcal{C}$ 且 $\mathrm{DT}^{h(n)}$(mon-$\mathcal{C}$) $\not\subseteq \mathcal{C}$。这说明探究 $\mathrm{DT}^{\mathcal{O}(\log n)}$(mon-$\mathcal{C}$) 与 $\mathcal{C}$ 的区别是有意义的；我们在下述定理中（对某些情况）回答了这一问题。

定理 1.2. 对于任意电路复杂度类 $\mathcal{C}$，只要 mon-$\mathrm{TC}^0 \subseteq$ mon-$\mathcal{C}$，就有 DT(mon-$\mathcal{C}$) $= \mathcal{C}$。

因此，特别地，DT(mon-TC^0) $= \mathrm{TC}^0$。对于 $\mathcal{C}$ 不包含 TC^0 的情况，我们的了解较少。我们首先论证，AC^0 中的所有函数都可以由高度为次线性的单调决策树计算。更具体地说，对于任意常数 r，当 $d(n) = \Omega\left(\frac{n}{\log^r n}\right)$ 时，$\mathrm{AC}^0 \subseteq \mathrm{DT}^{d(n)}$(mon-$\mathrm{AC}^0$)（定理 8.1）。自然会想到，次线性高度能否进一步降低。特别是，DT(mon-AC^0) 是否等于 AC^0。为此，我们利用 [18]，中的一种技术，首先证明 DT(mon-AC^0) 中的函数存在一个否定受限电路：

定理 1.3. 如果定义在 n 个变量上的布尔函数 f 属于 DT(mon-AC^0)，那么对于任意正数常数 $\epsilon \leq 1$，存在一个计算 f 的 AC^0 电路，其中包含 $\mathcal{O}(n^\epsilon)$ 个取反门。

备注 1.1. 根据 *[18]*, 定理 4.3，我们知道，否定函数 Neg：$\{0,1\}^n \to \{0,1\}^n$定义为 Neg(x) $= x \oplus 1^n$ 无法由常数深度电路和 $\mathcal{O}(\sqrt{n})$ 个取反计算得到。我们注意到，这并不能立即说明 DT(mon-AC^0) $\neq \mathrm{AC}^0$，因为 Neg 的输出并非单比特。

与定理 1.3形成鲜明对比的是，利用 [18] 可以推导出：如果 $f \in \mathrm{AC}^0$ 具有 $\mathsf{alt}(f) = \Omega(n)$ 的渐近复杂度，那么计算它的任意 AC^0 电路，对于某个常数，都必须至少包含 $\Omega(n^\epsilon)$ 个取反$\epsilon > 0$ （见定理 8.2）。因此，如果在取反数量方面实现渐近改进，就意味着 $\mathrm{DT}(mon\text{-}\mathrm{AC}^0) \neq \mathrm{AC}^0$。

在介绍这些主要结果的过程中，我们还指出，针对决策列表（定义见第 2 节）类似定义的函数类 DL(mon-$\mathcal{C}$) 恰好等于 DT(mon-$\mathcal{C}$)。我们针对随机决策树，以类似于 DT(mon-$\mathcal{C}$) 的方式定义 RDT(mon-$\mathcal{C}$)，并证明 RDT(mon-$\mathcal{C}$) $=$DT(mon-$\mathcal{C}$) $=$ DL(mon-$\mathcal{C}$) $= \mathcal{C}$ if mon-$\mathrm{TC}^0 \subseteq$ mon-$\mathcal{C}$，且 DL(mon-AC^0) $=$ DT(mon-AC^0) $\subseteq$RDT(mon-AC^0) $\subseteq \mathrm{AC}^0$。

[4]我们假设，在$\mathcal{C}$中，所有电路的规模都是多项式级的，并且至少存在一个具有 $\Omega(n)$ 交替数的函数。对于 AC^0,TC^0,NC^1 等复杂度类，这一假设成立。

2 预备知识

本节中，我们将定义基本术语和符号，并介绍主要的单调决策树复杂度度量。非确定性、随机化单调决策树以及其他变体的定义将在相应章节中给出。除非另有说明，本文讨论的布尔函数取自 $\{0,1\}^n$ 到 $\{0,1\}$。关于布尔电路及相关复杂度类的标准定义，请参阅 [15]。

单调性与交替数：对于 $x \neq y \in \{0,1\}^n$，如果 $\forall i \in [n], x_i \leq y_i$，我们称 $x \prec y$。定义在 $\{0,1\}^n$ 上的链 $\mathcal{X}$ 是一个序列 $\langle x^{(1)}, x^{(2)}, \ldots, x^{(\ell-1)}, x^{(\ell)} \rangle$，满足 $\forall i \in [\ell], x^{(i)} \in \{0,1\}^n$ 和 $x^{(1)} \prec x^{(2)} \prec \ldots \prec x^{(\ell)}$。函数 f 在链 $\mathcal{X}$ 上的交替数，记作 $alt(f, \mathcal{X})$，是序列 $\langle f(x^{(1)}) \ f(x^{(2)}) \ \ldots f(x^{(\ell)}) \rangle$ 中比特翻转（或交替）的次数。我们将 f 的交替数定义为 $\mathrm{alt}(f) := \max_{\text{链}\ x} alt(f, \mathcal{X})$。

如果对于所有 $x, y \in \{0,1\}^n, x \prec y \Rightarrow f$，都有 $x \leq f(y)$，我们就称一个布尔函数是单调的。若一个布尔电路中的所有门都在各自的输入上计算单调函数，则称该电路为单调电路。对于任意电路复杂度类 $\mathcal{C}$，我们将 mon-$\mathcal{C} \subseteq \mathcal{C}$ 定义为能够通过使用 $\mathcal{C}$ 中的单调电路计算的函数类。

阈值函数：对于 $x \in \{0,1\}^n$，我们用 $\mathrm{wt}(x)$ 表示 x 中 1 的个数。我们将 (k 阈值)函数定义为：若 $\mathrm{wt}(x) \geq k$，则 $\mathrm{Th}_k(x) = 1$；否则 $\mathrm{Th}_k(x) = 0$。

单调决策树与决策列表：现在我们介绍对决策树（及决策列表）模型的推广。在本文后续部分，我们将通过允许非自适应性、随机性、受限查询等方式，引入并研究更多变体。

定义 2.1（单调决策树）。单调决策树 T 是一棵有根有向二叉树。每个内部节点 v 都标记一个单调函数 $f_v : \{0,1\}^n \to \{0,1\}$，每个叶节点都标记为 0 或 1，e。每个内部节点有两条出边，一条标记为 0，另一条标记为 1。对于输入 $x \in \{0,1\}^n$，T 的一次计算是从根节点到某个叶节点 L 的路径；在每个内部节点 v 处，沿着标记与 $f_v(x)$ 的值相同的边前进。该路径到达的叶节点标记就是此次计算的输出。当且仅当对于每个输入 $x \in \{0,1\}^n$，T 的输出等于 $f(x)$ 时，单调决策树 T 才计算函数 $f : \{0,1\}^n \to \{0,1\}$。

f 的单调决策树复杂度是计算f的此类树的最小高度[5]。我们用 $\mathrm{DT}_m(f)$ 表示该值。

定义 2.2（单调决策列表）。单调决策列表模型记作$L = (f_1, c_1)(f_2, c_2) \ldots (f_k, c_k)$，它是一系列元组 (f_i, c_i)，其中每个 f_i 都是定义在 n个变量上的单调函数，而每个 c_i 都是布尔常数 0 或 1。这里，每个 (f_i, c_i) 都称为一个节点；f_i 该节点的查询

[5] $Height$ 始终指从根节点到任意叶节点的路径中内部节点的最大数量。

函数以及 c_i 该节点的值。最后一个查询 f_k 通常可以不失一般性地假设为常值函数 1。若输入 $x \in \{0,1\}^n$ 激活了[6] 节点 (f_i, c_i)，则称该输入满足 $f_i(x) = 1$且 $\forall j < i, f_j(x) = 0$。这里，$L$ 表示/计算如下布尔函数 f_L：$f_L(x) = c_i$，其中 $i \in [k]$ 是使得 x 激活 L的第 i^{th} 个节点的唯一索引。

布尔函数 f 的单调决策列表复杂度记为 $\mathrm{DL}_m(f)$，它是计算该函数的单调决策列表的最小规模（即节点数）。

文献中研究过一种决策列表：允许查询函数是变量的简单与，称为单调项决策列表 [13]。若允许查询函数为一般项（不一定是单调的），则称为项决策列表 [8]，而规模至多为 $\mathrm{poly}(n)$ 的 TDL 所计算的函数类记为 TDL。

2.1 MDL 的标准形式

通过标准论证，我们证明：当查询可以取自任意足够丰富的布尔函数类时，单调决策列表可以假定具有某些性质——尤其是函数集 mon-AC^0, mon-TC^0, mon-NC^1 等。

性质 1——交替常数：我们可以将任意计算布尔函数 f 的决策列表 $L = (f_1, c_1)(f_2, c_2)\ldots(f_k, c_k)$（其定义在 n 个变量上）转换为 $\tilde{L} = (f_1, \tilde{c_1})(f_2, \tilde{c_2})\ldots(f_k, \tilde{c_k})$，使其计算相同的函数，且常数 $\tilde{c_i}$ 在 0 和 1 之间交替。

这个思路很简单：将具有相同 c_i 的连续节点合并为一个节点，并对相应的查询执行或运算（例如，[2] 中的定理 3.1 就采用了这种做法）。

假设一段极大的节点序列 (f_i, c_i)、(f_{i+1}, c_{i+1}).....(f_j, c_j) 具有相同的常数值（即 $c_i = c_{i+1} \cdots c_j$）。我们可以用节点 $(f_i \vee f_{i+1} \cdots \vee f_j, c_i)$ 替换这整段节点。对所有常数值相同的极大连续节点应用此简化后，最终得到一个等价的、常数交替的单调决策列表。

为说明这一变换不会影响输出，我们论证如下简化成立。若 $(f_1, c)(f_2, c)$ 是决策列表中相邻的两个节点，则可以将它们替换为单个节点 $(f_1 \vee f_2, c)$，从而得到等价的决策列表。为此，我们注意到，若对于某个输入 x，f_1 和 f_2 都未通过（求值为 0），那么由于原来的两个节点都不会被激活，而新的节点 $(f_1 \vee f_2, c)$ 也不会被激活，因此这一替换不会改变决策列表返回的输出。另一方面，假设节点 (f_1, c) 是激活节点。这意味着在它之前的所有查询 f_1 的求值结果都是 0。由于这些节点保持不变，且 $f_1 \vee f_2$ 会通过 x，因此节点 $(f_1 \vee f_2, c)$ 就是激活节点，从而返回 c，与原决策列表返回的值相同。同样的

[6] 根据此定义，任意输入都恰好激活一个节点。

论证也适用于 (f_2, c) 是原决策列表中由 x 激活的节点的情况。

性质 2 - 前向触发：前向触发是指，对于任意输入 $x \in \{0,1\}^n$，如果决策列表中的某个查询通过（即计算结果为 1），那么其后所有查询也都会通过。

我们声称，决策列表 $\widetilde{L} = (g_1, c_1)(g_2, c_2)\ldots(g_k, c_k)$（其中 $g_i = \bigvee_{j=1}^{i} f_j$）表示 f，并具有前向触发性质。首先，注意到由于 $g_{i+1} \equiv g_i \vee f_{i+1}$，我们立即有 $\widetilde{\ }$，因此 $g_i \Rightarrow g_{i+1}$ 成立。现在，为了证明 L 等价于 L，假设在输入 x上，t^{th} 节点在 L中被激活。接着注意到，由于根据 t的定义，$f_t(x) = 1$，因此 $g_t(x) = \bigvee_{j=1}^{t} f_j(x) = 1$。对于 $i < t$，我们有 $g_i(x) = \bigvee_{j=1}^{i} f_j(x) = 0$，因为每个 f_j 对于 $j < t$ 都失败了，而 x 使 L的前 $t-1$ 个查询全部失败。因此， x 确实也会激活 $\widetilde{L}$ 的 t^{th} 节点，所以 $\widetilde{L}$ 在输入 x上的输出为 $c_t = f(x)$。

容易注意到，如果决策列表已经满足性质 1，那么确保性质 2 的过程不会破坏性质 1。也就是说，存在一个同时满足这两个性质的决策列表。上述标准形式适用于这样一种情形： f_i 属于一个对多项式个比特的或运算封闭的电路复杂度类，并且当 k 是 n 的多项式时，新的查询函数也属于该类（它们拥有属于该类的单调电路）。

回顾第2节中函数的交替数定义，我们给出以下布尔函数的刻画，该刻画最初由 [6]证明。

引理 2.1（交替性的刻画 [6]）。对于任意 f： $\{0,1\}^n \to \{0,1\}$ 存在$k = \mathrm{alt}(f)$ 个单调函数 $f_1, \ldots, f_k$，它们各自从 $\{0,1\}^n$ 映射到 $\{0,1\}$，使得

$$f(x) = \begin{cases} \oplus_{i=1}^{k} f_i(x) & , \text{if } f(0^n) = 0 \\ \neg \oplus_{i=1}^{k} f_i(x) & , \text{if } f(0^n) = 1. \end{cases}$$

3 我们的工具：布尔函数的单调分解

受前一节所述刻画的启发，我们如下定义布尔函数的单调分解。在分析单调决策树复杂度度量时，这一概念将很有帮助。

定义 3.1（布尔函数的单调分解）。对于任意布尔函数f： $\{0,1\}^n \to \{0,1\}$， f的单调分解是由单调函数$M = \{f_1, f_2, \ldots f_k\}$ 组成的最小集合，使得 $f = \oplus_{i \in [k]} f_i$——这里， k称为分解的长度。我们称分解中的每个 f_i 为一个单调分量。

我们考虑通过施加额外约束得到的两种单调分解变体：

蕴含性质：存在一种对 M 的排序，使得 $\forall i \in [k-1]$、 $f_i \Rightarrow f_{i+1}$ 成立。在这种

情况下，这些函数被称为 f 的分解的边界函数。我们将满足此性质的单调分解称为 f 的边界分解。

最优性：如果集合 M 也是 f 的规模最小的单调分解。也就是说，不存在由更少个单调函数组成的集合，其奇偶性等于给定函数 f。我们将满足这一性质的单调分解称为函数 f 的交替分解。

将布尔函数分解为单调函数奇偶性的证明 [6] 实际上蕴含了一个长度为交替数(f)（或交替数(f) $+1$，如果 $f(0^n) = 1$）的单调分解，该分解具有最优性和蕴含性质。他们的证明还表明，如果一个单调分解满足最优性，那么该分解的长度必然等于交替数(f) 或交替数(f) $+1$。因此，我们将具有最优性的分解称为交替分解。现在陈述以下引理（已在[6]中证明），以详细说明并证实我们所需的额外性质——其证明见附录A.1。

引理 3.1（单调分解引理）。对于任意布尔函数 $f: \{0,1\}^n \to \{0,1\}$，存在一个具有蕴含性质和最优性性质的单调分解，其长度为 alt(f)（若 $f(0^n) = 0$），且长度为 alt(f) $+1$（若 $f(0^n) = 1$）。

我们经常会用到这样一个事实：具有蕴含性质的单调分解对应于一个单调决策列表：

命题 3.1。设 k 是一个偶数，[7] 且 $f_1 \Rightarrow \cdots \Rightarrow f_k$ 是布尔函数。以下函数彼此等价：

- $f_1 \oplus f_2 \oplus \cdots \oplus f_k$,
- $(f_1, 0)(f_2, 1) \ldots (f_k, 1)(1, 0)$,
- $\overline{f_1} f_2 \vee \overline{f_3} f_4 \vee \cdots \vee \overline{f_{k-1}} f_k$,
- $(0 \vee \overline{f_1}) \wedge (f_2 \vee \overline{f_3}) \cdots \wedge (f_{k-2} \vee \overline{f_{k-1}}) \wedge (f_k \vee \overline{1})$.

证明。由于 f_i 的蕴含性质，对于任意输入 x，比特序列 $s := \langle f_1(x)\ f_2(x) \ldots, f_k(x) \rangle$ 是有序的（从 0 到 1）。上述四个函数都计算序列 s 中 1 的个数是否为奇数。为此，首先检查当 $s = 1^k$时，所有函数的值都为 0。每当我们将 s 的最后一个 1 比特翻转为 0 时，上述四个函数的函数值都会在 0 和 1 之间交替变化。

□

[7]这只是一个次要限制，因为我们可以在单调分解或 MDL 前后添加常值函数。

3.1 特殊情形下交替分解的唯一性

对于给定的函数，可能存在多个满足蕴含性质和最优性性质的单调分解。例如，考虑任意偶数 $n > 2$ 对应的函数 $f = (x = 0^{n/2}1^{n/2})$。当且仅当输入为 $0^{n/2}1^{n/2}$ 时，该函数的值为 1。显然，此函数的交替数为 2。我们将给出该函数的两种不同分解。一种是 $f(x) = \mathrm{Th}^{n/2}(x) \oplus (\mathrm{Th}^{n/2}(x) \wedge (x \neq 0^{n/2}1^{n/2}))$。另一种分解是 $f(x) = \mathrm{Th}^{n/2+1}(x) \oplus (\mathrm{Th}^{n/2+1}(x) \vee (x = 0^{n/2}1^{n/2}))$。不难验证，这两种单调分解都满足附加性质。不过，在一种特殊情况下，我们确实有唯一的、满足蕴含性质和最优性性质的单调分解:

命题 3.2. 如果布尔函数 f 在布尔超立方体中所有最大长度链上都呈现一致交替，那么 f 具有满足蕴含性质的唯一交替分解。

证明. 假设 f 有两个满足蕴含性质的交替分解。将它们记为 $\{f_1, f_2, \dots f_k\}$ 和 $\{f'_1, f'_2, \dots f'_k\}$，使得 $f_1 \Rightarrow f_2 \Rightarrow \dots f_{k-1} \Rightarrow f_k$ 和$f'_1 \Rightarrow f'_2 \Rightarrow \dots f'_{k-1} \Rightarrow f'_k$，且 $k = \mathrm{alt}(f)$（假设 $f(0^n) = 0$）。我们将通过反证说明，这些函数的集合必定相同，即对所有 i，$f_i = f'_i$。考虑满足存在某个 $x \in \{0,1\}^n$ 使得 $f_i(x) \neq f'_i(x)$ 的最大 i。不失一般性，设 $f_i(x) = 1$，且$f'_i(x) = 0$。固定这样一个 x 和一条包含 x 的链 $0^n = x^{(0)} \prec x^{(1)} \prec x^{(2)} \prec \cdots \prec x^{(n)} = 1^n$，使得上述链中位于 x 之前的所有输入 y 都满足 $f_i(y) = 0$（因此 $f_{i-1}(y) = 0$ 也成立）。利用最优性性质，我们得到 $f = f_1 \oplus f_2 \cdots \oplus f_k = f'_1 \oplus f'_2 \cdots \oplus f'_k$。

如果 $i = 1$，那么这两个分解在输入 x 上的结果不同，因为对于 $i > 1$，$f_i = f'_i$，这构成矛盾。

如果 $i > 1$，我们必然有 $f_{i-1}(x) = 1$，否则 $1 = f_1 \oplus \cdots \oplus f_i = f'_1 \oplus \cdots \oplus f'_i = 0$（因为 i 是满足 $f_i \neq f'_i$ 的最大索引，且这是由蕴含性质得出的）。因此，在我们所考虑的链中，当输入从紧邻 x 之前的输入变为 x 时，f_i 和 f_{i-1}的取值都会从 0 变为 1。这与该链只有 k 次交替的事实矛盾。

□

3.2 TC^0 及更高类的单调分量约束

本节继续研究函数 f，并对其施加复杂度约束。一个自然的问题是，在 f 的单调分解中，其单调分量在电路复杂度类方面是否必然比 f 更难。我们首先针对包含阈值函数单调电路的类回答这个问题。在这种情况下，我们证明总能找到一种单调分解，使得分量函数属于单-$\mathcal{C}$。

引理 3.2. 若 $\text{mon-}\mathrm{TC}^0 \subseteq \text{mon-}\mathcal{C}$，那么对于任何由类 $\mathcal{C}$ 中的电路计算的 f，都存在一个具有蕴含性质的单调分解 $f_1 \oplus f_2 \oplus \dots \oplus f_{2n+1}$，使得每个 f_i 都属于 mon-$\mathcal{C}$。

证明。回顾一下，Th_k 表示函数 $\mathsf{Th}_k(x) = 1$ 当且仅当 x 中 1 的个数（记作 $\mathsf{wt}(x)$）至少为 k。我们直接写出这个分解。对于每个 $1 \le i \le 2n+1$，若 i 为 $2k+1$，则 $f_i(x) = \mathsf{Th}_{k+1}(x) \vee (\mathsf{Th}_k(x) \wedge f(x))$；否则 $f_i(x) = \mathsf{Th}_k(x)$，当 i 为 $2k$ 时。
正确性：令 $w = \mathsf{wt}(x)$。对于任意 $x \in \{0,1\}^n$：

对于 $i < w$：我们有 $\mathsf{Th}_i(x) = 1$ 且 $\mathsf{Th}_{i+1}(x) \vee (\mathsf{Th}_i(x) \wedge f(x)) = 1$。

对于 $i > w$：我们有 $\mathsf{Th}_i(x) = 0$ 且 $\mathsf{Th}_{i+1}(x) \vee (\mathsf{Th}_i(x) \wedge f(x)) = 0$。

对于 $i = w$：我们有 $\mathsf{Th}_i(x) = 1$ 且 $\mathsf{Th}_{i+1}(x) \vee (\mathsf{Th}_i(x) \wedge f(x)) = f(x)$

利用这一点，我们可以如下计算边界函数：

$$\begin{aligned} \forall i, 1 \le i \le 2w: \ f_i(x) &= 1 \text{ and } \forall i, 2w+2 \le i \le 2n+1: \ f_i(x) = 0 \\ f_{2w+1}(x) &= \mathsf{Th}_{w+1}(x) \vee (\mathsf{Th}_w(x) \wedge f(x)) = f(x) \end{aligned}$$

观察上述计算可知，对于 f_i，蕴含性质成立，因为 $f_{2n+1} \Rightarrow f_{2n} \Rightarrow \cdots \Rightarrow f_2 \Rightarrow f_1$。表达式 $f_1(x) \oplus f_2(x) \cdots \oplus f_{2n+1}(x)$ 的计算结果为 $1 \oplus 1 \ldots 1 \oplus f(x) \oplus 0 \cdots \oplus 0$，上述表达式中 $f(x)$ 之前 1 的个数为 $2w$（这是偶数）。因此，对于给定的输入 x，该分解的计算结果为 $f(x)$。

f_i 的复杂度界：现在我们将证明这些查询函数确实属于单调 $\mathcal{C}$。偶数 i 对应的 f_i 是阈值函数，因此满足此性质。下面我们证明，$f_{2k+1} = \mathsf{Th}_{k+1} \vee (\mathsf{Th}_k \wedge f)$ 在 $\mathcal{C}$ 电路中具有一个规模为 $0 \le k \le n$ 的单调电路。

设 C 是一个计算 f 的 $\mathcal{C}$ 电路。首先，利用 $\neg(\mathsf{Th}_\ell(a_1, a_2, \ldots a_m)) = \mathsf{Th}_{m-\ell+1}(\neg a_1, \neg a_2, \ldots \neg a_m)$ 这一事实，将所有否定门沿 C 下推至输入变量。这样做只会使 C 的规模增加多项式级，且深度保持不变。在所得电路中，按如下方式进一步消去变量上的取反：如果出现 x_j，就将其替换为 $\mathsf{Th}_k(\{x_1, x_2, \ldots x_n\} \setminus \{x_j\})$。将最终得到的电路记为 C'。注意，C' 不含任何否定门，因此它是 $\mathcal{C}$ 上的单调电路。我们将论证电路 $C'' = \mathsf{Th}_{k+1} \vee (\mathsf{Th}_k \wedge C')$ 计算的是 f_{2k+1}。可以看出，对于重量小于 k 的输入，C'' 输出 0；对于重量大于 k 的输入，则输出 1，这恰好与 f_{2k+1} 在这些输入上的计算结果一致。因此，只需证明对于重量恰好等于 k 的输入，C'' 能正确输出 $f_{2k+1}(x) = f(x)$。为此，注意到当 $\mathsf{wt}(x) = k$ 时，$x_j = \mathsf{Th}_k(\{x_1, x_2, , \ldots x_n\} \setminus \{x_j\})$ 的最终变换成立。

—

□

根据上述引理，对于任意 $f \in \mathsf{TC}^0$，我们都有一个由 $2n+1$ 单调 TC^0函数组成的分解。然而，这种单调分解可能远未达到最优性。如果该函数在所有长度为 $n+1$ 的链上都具有一致交替数，那么可以将其改进为 $\mathsf{alt}(f)$，同时保持单调分量的复杂度不超过 TC^0, ，如下所示。

TC^0 中“特殊”函数的单调分解：

命题 3.3. 如果 $f \in \mathsf{TC}^0$ 在所有长度为 $n+1$ 的链上都具有一致交替，记作 $\mathsf{alt}(f) = k$，那么 f 存在一个长度为 k 或 $k+1$ 的单调分解，其中所有分量都属于 TC^0。

证明。 值得回顾的是，当所有长度为$n+1$ 的链具有相同的交替数时，分解是唯一的（命题 3.2）。我们只在 $f(0^n) = 0$（分解为 k 单调 TC^0 分量）的情况下给出证明。在另一种情况下，我们只需对补函数的分解取否定（所得分解仍属于 TC^0 且具有一致交替数）。

设 $f \equiv f_1 \oplus f_2 \oplus \cdots \oplus f_k$，其中我们有 $k = \mathsf{alt}(f)$ 和 $\forall i \in [k-1]: \quad f_i \Rightarrow f_{i+1}$。我们利用 f_i 的以下结构，构造一个计算它的 TC^0 电路。这个观察源于 f的一致交替特性。

观察 3.1。对于任意 $1 \leq i \leq k$， $f_i(x) = 1$ 当且仅当存在一条从 0^n 到 x 的链，且至少有 $k-i+1$ 次交替。

由于所有从 0^n 到 x、长度为 $\mathsf{wt}(x)+1$ 的链都有相同的交替次数，我们可以任取这样一条链来确定 $f_i(x)$ 的值。我们将考虑通过反复将最左侧的 1 变为 0 而得到的输入链。

对于输入 $x \in \{0,1\}^n$，我们归纳地定义 n 个新字符串 $y^{(i)}$。字符串 $y^{(1)}$ 是通过将 x 中最左侧的 1 变为 0 得到的；类似地，对于任何其他 $i > 1$，我们将 $y^{(i-1)}$ 中最左侧的 1 变为 0，从而得到 $y^{(i)}$。如果 x 中 1 的数量小于 i，那么我们定义 $y^{(i)}$ 为 0^n。现在我们将说明如何在 TC^0 中计算 $y^{(i)}$。由于 $\textsc{BitCount} \in \mathsf{TC}^0$,，我们可以在 TC^0 中得到所有前缀和 $p^{(i)} \in \{0,1\}^{\log n} = \sum_{j=1}^{j=i} x_j$。于是，$y^{(i)}$的 j^{th} 比特，即 $y_j^{(i)} := x_j^{(i)} \wedge (p^{(j)} > i)$。这是因为，对于 x 的前 i 1 个比特，前缀和 $p^{(i)}$ 至多为 i，而对于所有其他比特，它都大于 i。此外，这可以在 TC^0 中实现，因为我们知道关系 $>$ 属于 AC^0。

得到 $y^{(i)}$ 后，我们构造长度为 $n+1$ 的布尔序列 s， $s := f(x).f(y_1).f(y_2).\ldots f(y_n)$。由于我们知道 $f \in \mathsf{TC}^0$，且每个 $y^{(i)}$ 都可以在 TC^0 中并行计算，因此这也可以在 TC^0 中完成。注意，序列 $\langle y^{(n)}、\ y^{(n-1)}、\ \ldots y^{(1)}、\ x\rangle$ 在某个位置之前表示相同的输入 0^n，并从下一个位置开始表示一条从 0^n 到 x 的链。因此，根据我们的观察以及 $f(0^n) = 0$ 这一事实，我们注意到， $f_i(x) = 1$ 当且仅当 $alternations(s) \geq i$。现在，为了计算序列 s 中的交替次数，我们比较 s 中每两个相邻的比特，并检查其变化次数是否至少为 i。也就是说， $f_i \equiv \mathsf{Th}^{k-i+1}(s_1 \oplus s_2, s_2 \oplus s_3, \ldots, s_n \oplus s_{n+1})$。显然，这也可以在 TC^0 中完成，因为位奇偶校验和阈值门都属于 TC^0。

需要指出的是，我们只给出了 TC^0 中单调函数的分解。我们不知道这些函数是否具有单调 TC^0 电路。当然，另一个局限是，这种方法只适用于具有一致交替数的函数。

□

4 确定性自适应单调决策树

我们首先研究的模型是确定性自适应单调决策树，以及引言中定义的相关复杂度度量，如 DT_m 和 DL_m 。

4.1 单调决策树与单调决策列表

本节讨论单调决策树与单调决策列表的复杂度之间的关系。特别地，我们证明以下定理：

定理 4.1. 对于任意布尔函数 f， $\mathsf{DT}_m(f) = \lceil \log \mathsf{DL}_m(f) \rceil$ [8]。

证明。($\geq$)：为证明 $\mathsf{DL}_m(f) \leq 2^{\mathsf{DT}_m(f)}$，注意到只需证明：给定一棵单调决策树，我们可以构造出一份规模相同的单调决策列表。由于最优树的叶节点数至多为 $2^{\mathsf{DT}_m(f)}$，因此可得该不等式。现在来证明这一关系本身。它可由 Blum 给出的一种已知构造推出 [7]。由于我们还需要该构造适用于单调查询，因此在下面的引理中给出一个自成一体的证明。

引理 4.1. 设 T 是一棵规模为 k 、计算定义在 n个变量上的函数 f 的单调决策树。那么存在一份规模为 k 、计算 f的单调决策列表。

证明。对于 T 中的每个叶节点 ℓ，将从源点/根节点到 ℓ 的路径表示为字符串 $s_\ell \in \{0,1\}^*$，其定义为：当且仅当路径上的 i^{th} 条边标记为正时， $s_\ell \in \{0,1\}^* = s_\ell[i] = 1$ 。

令 $S = s_1$、s_2、.....、s_k 为所有此类字符串按字典序从大到小的排序。这等价于将树的叶节点从右向左排序（按照约定，在决策树的每个查询节点处，左边对应 0，右边对应 1）。

我们声称，以下单调决策列表 L 计算 f：

$$L = (t_1, label(\ell_1))(t_2, label(\ell_2)) \dots (t_k, label(\ell_k)),$$

其中， $label(\ell_i) \in \{0,1\}$ 表示叶节点 ℓ_i 的标签，而 t_i 是由沿着从 根节点到 T 的路径上经过的所有查询 的合取得到的函数，至 ℓ_i （见图2）。

为证明这一点，令输入 x 激活 L 的 i^{th} 节点。我们有 $t_i(x) = 1$。现在只需证明， T 对 x 的计算会到达叶节点 ℓ_i ，因此输出相同的值$label(\ell_i)$ 。

用反证法证明这一点，假设结论不成立；也就是说，设决策树在输入 x 时到达的叶节点为 $\ell_j \neq \ell_i$。对应路径（字符串）s_j 和 s_i 首次不同的位置是第 k 位。这样的位置必然存在，否则一个字符串就是另一个的前缀；但这是不可能的，因为这些二进制字符串“编码”了树中从根节点到叶节点的路径。

[8]使用相同的证明，对于具有适当闭包性质的任意电路复杂度类，我们还可以得到 $\mathcal{C}$ $\mathsf{DT}(mon\text{-}\mathcal{C}) = \mathsf{DL}(mon\text{-}\mathcal{C})$——这一结论将在第 8 节中使用。

情形（1）— $s_j[k] = 0$ 和 $s_i[k] = 1$: 由于这些字符串直到第（$k-1$）th 位都相同，因此决策树中对应的路径也相同。设 ℓ_i 和 ℓ_j 的最近公共祖先节点标记为一个（单调）查询 f_1（不失一般性）。注意，由于 $s_i[k] = 1$，它使 f_1 成为通往 ℓ_i 路径上的一个已通过查询，因此 $f_1 \in t_i$（即，按 t_i的定义， t_i 是 f_1 与其他函数的与）。由于我们有 t_i（x）$= 1$，这意味着 f_1（x）$= 1$，也就是说 $s_j[k] = 1$，矛盾。

情况（2）- $s_j[k] = 1$ 和 $s_i[k] = 0$: 这意味着 $s_j > s_i$。注意，由于决策树中的计算在 x 上到达 ℓ_j，且 t_j 是通向 ℓ_j的路径上已通过查询的合取，因此必有 $t_j(x) = 1$。又由于 $s_j > s_i$，查询 t_j 会先于 t_i 出现在L中。因此，节点（$t_i, label(\ell_i)$）不会在 x上被激活，这与i 本身的定义矛盾。

因此，引理的证明得证。

□

（$\leq$）：现在，为证明 $\mathrm{DT}_m(f)$ 至多为 $\lceil \log \mathrm{DL}_m(f) \rceil$，我们构造一棵高度为 $\lceil \log k \rceil$ 的单调决策树，给定一个等价的、长度为 k的单调决策列表。证明见下一个引理。

引理 4.2。设 $L = (f_1, c_1)(f_2, c_2) \ldots (f_{k-1}, c_{k-1})(1, c_k)$ 是布尔函数 f 在 n 个变量上的一个单调决策列表。那么存在一棵高度为 $\lceil \log k \rceil$、计算 f 的单调决策树 T。

证明。 不失一般性，我们假设决策列表 L 处于标准形式（第 2.1 节）。对于输入 $x \in \{0,1\}^n$，恰好存在一个节点，使其对应的查询及其右侧的所有查询均通过，而其左侧的所有查询均不通过。一种自然的方法是通过查询相应的函数，算法化地搜索这个切换点（即激活节点）；具体做法是进行二分搜索，并返回 c_i 搜索所得索引对应的 T，从而构造一个关于 f 的决策树。

在 T 的根节点处，我们查询 $f_{\lceil (1+k)/2 \rceil}$。如果结果为零（沿树向左），这意味着激活节点位于它的右侧；否则（沿树向右），激活节点位于它的左侧（或就是该节点）。这样，我们便可递归地构造 T（在左分支使用 L 的右半部分，在右分支使用 L 的左半部分）。为了获得 T 的叶节点标签，我们从右向左写下 c_1、$c_2 \ldots c_k$。

例如，若 $L = (f_1, c_1)(f_2, c_2)(f_3, c_3)(f_4, c_4)(f_5, c_5)(f_6, c_6)(f_7, c_7)(1, c_8)$，则构造出的树 T 如图1所示。可以看出， T 的高度为 $\lceil \log k \rceil$，因为在每次查询中，剩余决策列表的节点几乎平均分布在左右两侧。最后，我们注意到， T 中的查询与 L中的查询相同，因此是单调的。

□

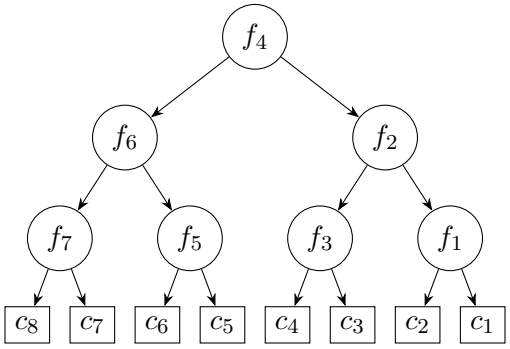

图1：与 MDL 对应的单调决策树为 $(f_1, c_1)(f_2, c_2) \ldots (f_7, c_7)(1, c_8)$

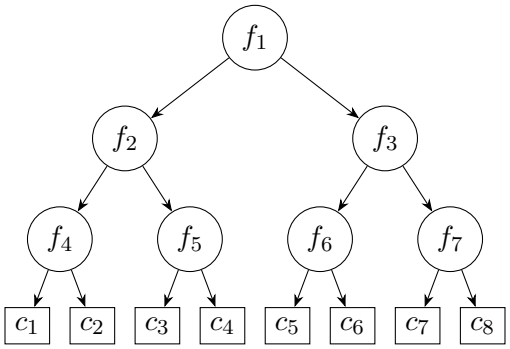

图 2：与上述单调决策树对应的 MDL 为 $(f_1.f_3.f_7, c_8)(f_1.f_3, c_7)(f_1.f_6, c_6)(f_1, c_5) \ldots (1, c_1)$

至此，关于单调决策列表复杂度与单调决策树复杂度之间关系的证明就完成了。

□

4.2 刻画自适应决策树高度

现在我们将证明本节的主要定理，该定理用 alt(f) 刻画 $\mathrm{DT}_m(f)$（顺带也刻画 $\mathrm{DL}_m(f)$）。

定理 4.2. 对任意布尔函数 f，$\mathrm{DT}_m(f) = \lceil \log(\mathrm{alt}(f) + 1) \rceil$。

证明。只需证明$\mathrm{DL}_m(f) = \mathsf{alt}(f) + 1$，这是由定理 4.1得出的。

$\mathrm{DL}_m(f) \leq \mathrm{alt}(f) + 1$: 首先，假设 $f(0^n) = 0$。由引理 3.1 可知，存在 $k = \mathrm{alt}(f)$ 个单调函数，使得 $f = f_1 \oplus f_2 \oplus \cdots \oplus f_k$，且 $\forall i, f_i \Rightarrow f_{i+1}$。容易验证，当 k 为偶数或奇数时，单调决策列表 $(f_1, 0)(f_2, 1)(f_3, 0)(f_4, 1) \ldots (f_k, 1)(1, 0)$ 或 $(f_1, 1)(f_2, 0)(f_3, 1)(f_4, 0) \ldots (f_k, 1)(1, 0)$ 分别计算 f。另一方面，若 $f(0^n) = 1$，则有 $f = f_1 \oplus f_2 \oplus \cdots \oplus f_k \oplus 1$，从而得到单调决策列表 $(f_1, 1)(f_2, 0) \ldots (1, 1)$ 或 $(f_1, 0)(f_2, 1) \ldots (1, 1)$；当 k 为偶数或奇数时，它们分别计算 f。

$\mathrm{DL}_m(f) \geq \mathrm{alt}(f) + 1$: 我们断言，如果一个定义在 n 个变量上的布尔函数 f 可以由长度为 ℓ 的单调决策列表 $L = (f_1, c_1)(f_2, c_2) \ldots (f_\ell, c_\ell)$ 计算，则 $\mathrm{alt}(f) \leq \ell - 1$。为证明这一点，只需说明，对于布尔超立方体中的任意一条链 $x^{(1)} \prec x^{(2)} \prec \ldots \prec x^{(s)}$，其中 $1 \leq s \leq n+1$；函数 f 沿该链的交替次数至多为 $\ell - 1$。

考虑一个长度为 s 的序列 S，其中对于 $1 \leq i \leq s$，整数 $S[i]$ 是将 $x^{(i)}$ 输入 L 时被激活节点的索引。因此，对于每个 i，都有 $1 \leq S[i] \leq \ell$。根据激活节点的定义，注意到对于任意 $1 \leq i < s$, $f_{S[i]}$，有 $x^{(i)} = 1$，这也意味着 $f_{S[i]}(x^{(i+1)}) = 1$，因为 $x^{(i)} \prec x^{(i+1)}$。这说明，$x^{(i+1)}$ 激活的节点不可能排在 $f_{S[i]}$ 之后。也就是说，对于所有 $1 \leq i < s$，都有 $S[i+1] \leq S[i]$。如果链中两个相邻元素激活了同一个节点，那么 L 在这些赋值上的输出相同，因此链上的该位置不会发生交替。因此，函数 f 沿该链的交替次数至多为满足 $S[i] \neq S[i+1]$ 的数对 $(S[i], S[i+1])$ 的数量。由于 $1 \leq S[i] \leq \ell$，且对于所有 i 都有 $S[i] \geq S[i+1]$，我们得到 $\mathrm{alt}(f) \leq \ell - 1$。

□

4.3 从否定受限电路构造自适应MDT s

上述定理根据布尔函数的交替数 alt(f)，给出了对决策树高度的刻画。Markov [17], 的经典结果表明，任意布尔函数都可以

由至多使用 $\lceil \log(\mathrm{alt}(f)+1) \rceil$ 个取反门的布尔电路计算。由于电路中的取反门数量可能在对数意义下更少，下面的结果很有趣。

定理 4.3. 设 f 是由使用 k 次取反的电路 C 计算的布尔函数。那么存在一棵高度为 $k+1$、计算 f 的单调决策树。

证明。 将 C 中最底层的否定门 g。将 g 的输入视为由一个单调电路（记作 C_+）计算。我们的单调决策树的第一个查询是由 C_+ 计算的函数。设 C_0 和 C_1 是分别将[9] g 替换为常数 0 和 1 后得到的电路。注意，这些电路含有 $k-1$ 个否定门。在决策树的 $C_+(x)=0$ 分支上，我们使用 C_1；另一侧则使用 C_0 代替 C，以此获得决策树第二层的查询。我们对 C 不断应用同一原则，直到没有剩余的取反，此时即可直接查询整个函数并得到返回值。在树的任意分支上，每次查询都会使树的高度增加 1，同时使剩余电路中的取反数量减少 1。因此，所构造的单调决策树的高度为 $k+1$。

□

5 确定性非自适应单调决策树

我们建立非自适应单调决策树与交替数之间的关系：

定理 5.1. 对于任意布尔函数，$\mathrm{alt}(f)=k$ 当且仅当 f 可以由高度为 k 的非自适应单调决策树计算。

证明。 概述一下这里采用的思路——对于正向推论，我们使用引理 3.1 来设计决策树。对于反向推论，由于该树是非自适应的，决策树每一层的查询函数都相同。利用这一点，我们论证任意链相对于 k 的交替数至多为 f。更具体地说，

($\Rightarrow$)：假设 $\mathrm{alt}(f)=k$。应用引理 3.1，我们如下描述高度为 k 的非自适应单调决策树。在第 i 层（根节点为第 1 层），所有节点都查询函数值f_i。标记叶节点时，只需用从树根到该叶节点的路径上各次查询结果的奇偶性来标记该叶节点。这样，根据定义，该树是非自适应的；并且在每条路径上，所有 $f_i(x)$ 的值都是已知的，因此可以按照上述方法计算 $f(x)$ 的值。

($\Leftarrow$)：设 f 由高度为 k的决策树 T 计算。由于 T 是非自适应的，因此在某一高度进行的查询所用的函数独立于之前的查询。设在高度i 进行查询的函数为 f_i。由于 T 是非自适应的，它也是一棵完全二叉树。用字符串 $s=s_1\ldots s_k\in\{0,1\}^k$表示二叉树的每个叶节点。字符串 s可以看作对 $\Leftarrow$进行查询的结果

[9]在需要 g 输出时，分别使用 0 和 1

。也就是说，要到达 T的某个叶节点（由 s表示），查询函数 f_1、f_2、$\ldots f_k$必须分别求得 s_1、s_2、$\ldots s_k$。

考虑任意一条从 $\mathcal{X} = x^{(1)} \prec x^{(2)} \prec \ldots x^{(\ell)}$到 f的输入链，并令相应的二进制分别为。我们将论证，这条链上的交替次数至多为 k。考虑沿链移动时会发生什么（例如，从 $x^{(j)}$到 $x^{(j+1)}$）。假设对于输入 $x^{(j)}$和 $x^{(j+1)}$，T分别计算出由 k比特字符串 $s^{(j)}$和 $s^{(j+1)}$表示的叶节点。由于 f_i是单调函数且 $x^{(j)} \prec x^{(j+1)}$，对于所有 i，都有 $f_i(x^{(j)}) \leq f_i(x^{(j+1)})$。因此，$s^{(j)} \prec s^{(j+1)}$或 $s^{(j)} = s^{(j+1)}$。由于 f的值完全由 f_i的值决定，若 $s^{(j)} = s^{(j+1)}$，则 $f(x^{(j)}) = f(x^{(j+1)})$。因此，$f$沿 $\mathcal{X}$的交替次数至多为字符串 $s^{(1)} \prec s^{(2)} \prec \ldots$的数量，而该数量至多为每个字符串的长度，即 k。

□

上述定理以及定理 4.2 完成了引言中定理 1.1 的证明。

定理 **1.1**。 对于任意布尔函数 f，$\mathrm{DT}_m(f) = \lceil \log(\mathrm{alt}(f) + 1) \rceil$，且 $\mathrm{DT}_m^{na}(f) = \mathrm{alt}(f)$.

5.1 单调证书复杂度

现在，我们通过对该函数证书复杂度的推广，讨论非自适应单调决策树复杂度的一种刻画。

定义 5.1（单调证书复杂度）。对于布尔函数 f的一个输入 $x \in \{0,1\}^n$，若一组定义在 n 个变量上的单调布尔函数 $S_x = \{f_1, f_2 \ldots f_k\}$ 满足：对于任意输入 $y \in \{0,1\}^n$，都有 $[\forall_{i=1}^k f_i(y) = f_i(x)] \Rightarrow [f(y) = f(x)]$，则称其为一个单调证书（集合）。$x$的单调证书复杂度记作 $\mathsf{C}_m(f,x)$，定义为 x的单调证书 S_x 的最小规模 $|S_x|$。函数 f 本身的单调证书复杂度定义为 $\mathsf{C}_m(f) := max_x\{\mathsf{C}_m(f,x)\}$。

有趣的是，对于任意函数f，单调证书集的规模都有一个常数上界。我们证明，$\mathsf{C}_m(f)$ 至多为 2。

命题 5.1. $\mathsf{C}_m(f) \leq 2$。

证明。对于任意输入 x，令 I_x 为 x中所有取值为 1 的变量组成的集合，令 J_x 为其余索引组成的集合。集合 $S_x = \{f_1 := \bigwedge I_x, f_2 := \bigvee J_x\}$ 是输入 x的一个有效单调证书。为证明这确实是一个证书，我们论证 $[(f_1(y) = f_1(x)) \wedge (f_2(y) = f_2(x))] \Rightarrow [f(y) = f(x)]$。首先，注意根据 f_i 的定义，$f_1(x) = 1$ 且 $f_2(x) = 0$。现在假设上述目标蕴含式的左侧为真。也就是说，$f_1(y) = 1$ 且 $f_2(y) = 0$。因此，$\bigwedge I_x(y) = 1$ 且$\bigvee J_x(y) = 0$，这意味着 $I_y \subseteq I_x$ 且 $J_y \subseteq J_x$。但注意，(I_y, J_y)（以及 (I_x, J_x)）构成变量全集的一个划分。因此，必有 $I_x = I_y$ 且 $J_x = J_y$，即 $x = y$。于是，所需蕴含式的右侧显然成立。

□

如果要求所有输入使用相同的单调证书集，我们将这种度量称为函数 f的非自适应单调证书复杂度，记作 $\mathsf{C}_m^{na}(f)$。通过一个简单的论证，我们有：

命题 5.2. $\mathsf{C}_m^{na}(f) = \mathsf{DT}_m^{na}(f) = \mathsf{alt}(f)$。

证明。($\leq$) 令 $h = \mathsf{DT}_m^{na}(f)$ 表示计算定义在 n 个变量上的布尔函数 f 的非自适应单调决策树（记作 T）的最小高度。从根节点到任意叶节点的查询函数依次为 f_1、$f_2 \dots f_h$，其中每个 f_i 都是定义在 n 个变量上的单调函数。我们断言，对于任意输入，$S := \{f_1, f_2, \dots f_h\}$ 都是一个单调证书。于是立即得到 $\mathsf{C}_m^{na}(f) \leq |S| = h = \mathsf{DT}_m^{na}(f)$。为观察到 S 确实是一个证书集，考虑两个输入 x 和 y，它们使 S 中的所有函数都取相同的值。那么在树 T 中，这两个输入都会到达同一个叶节点，因此返回相同的值。我们得到 $[\forall_{i=1}^{h}\ f_i(y) = f_i(x)] \Rightarrow [f(y) = f(x)]$，因此对于任意输入，$S$ 都是一个单调证书。

($\geq$) 鉴于 $S = \{f_1, f_2 \dots f_k\}$ 是一个单调证书集，我们的目标是为 f 设计一棵高度为 k的非自适应单调决策树 T。我们的思路是将 S 中的相同函数用作 T中的查询。我们固定 S 的一个任意排序，并在所有路径中按相同顺序查询这些函数。为了确定 T叶节点的标签，我们将叶节点 ℓ 的标签固定为 $f(x_\ell)$，其中x_ℓ 是任意一个输入，它在 T的计算过程中到达 ℓ。如果不存在这样的 x_ℓ，则可以任意标记 ℓ。设某个输入 x 到达 T 中的叶节点 ℓ。由于 x_ℓ 也到达 ℓ，因此 x 和 x_ℓ在 S中的所有函数上的计算结果都必须相同。根据 S的证书性质，这意味着 $f(x) = f(x_\ell)$，这正是 T在输入 x时返回的结果。

□

6 非确定性单调决策树

受布尔函数的非确定性决策树和证书复杂度的定义启发，我们也研究单调决策树的非确定性变体。我们将非确定性单调决策树定义为一种树：每个内部节点可以有一条或多条出边，树中的每条边都标记着一个单调函数或一个单调函数的否定，而叶节点标记为 0 或 1。如果存在一条从根节点到标记为 1 的叶节点的路径，且路径上所有边的标签函数取值均为 1，则称该输入被接受。

6.1 两种等价定义

我们定义非确定性单调决策树的两种变体。第一种变体将在本节后续部分中使用。

- Model1：一种树，其中每个内部节点可以有任意数量的出边，每条边都标记有一个单调函数或单调函数的否定。叶节点标记为 0 或 1。

如果存在一条从树的根节点到任意标记为 1 的叶节点的路径，且该路径上的所有边都处于激活状态，则称输入 x 被树接受（等同于在 x上输出 1）。这里，当 $f_i(x) = 1$ 时，我们称标记为 f_i 的边在 x 上处于激活状态。

- Model2：一种树，其中每个内部节点可以有任意数量的出边，每条边都标记为 0 或 1。叶节点照常标记为 0 或 1。除了这些标签外，内部节点还标记有单调函数。

  如果存在一条从树的根节点到任意一个标记为 1 的叶节点的路径，且路径上的所有边都处于激活状态，则称输入 x 被树接受（等同于在 x 上输出 1）。这里，对于从标记为 f_i 的节点出发的边 e，如果 e 的标签等于 $f_i(x)$ 的值，就称该边在输入 x 上处于激活状态。

如果一棵树 T（可以是 Model1 或 Model2 类型）计算布尔函数 f：$\{0,1\}^n \to \{0,1\}$，则称该树计算该函数，当且仅当对于所有 $x \in \{0,1\}^n$，T 在 x 上的输出为 $f(x)$。

可以证明，上述定义的模型是等价的。也就是说，我们证明：如果一个布尔函数 f：$\{0,1\}^n \to \{0,1\}$ 可以由一棵类型为 Model2 的树计算，那么也存在一棵高度相同（相差至多一个常数因子）、计算 f 的类型为 Model1 的树；反之亦然。

为证明第一部分，设 T 是一棵计算 f 的 Model2 树。我们只需按如下方式重新标记 T，即可得到一棵仍然计算 f 的 Model1 树。在每个标记为 f_i的内部节点处：如果一条出边标记为 0，就将其标签替换为 $\overline{f_i}$；如果一条出边标记为 1，就将其标签替换为 f_i。最后，移除所有内部节点上的标签，保留叶节点的标签。这样，我们就得到一棵树 T'。注意，对于同一个输入，当且仅当 T' 中对应的边是激活的，T 中的边才是激活的。因此，在 T 中激活的任何路径，在 T' 中仍然保持激活，这意味着 $T \equiv T' \equiv f$。

反方向由以下引理证明。

引理 6.1. 如果布尔函数 f： $\{0,1\}^n \to \{0,1\}$ 由类型为 Model1 的树 T 计算，那么也存在一棵类型为 Model2 的树 T'，其高度是 T 的两倍，并计算 f。

证明。为了构造 T'，我们从上到下对 T 的所有边执行以下操作。设 e 是 T 中的一条出边，它从标记为 f_i 的节点 i（可以是单调函数或单调函数的否定）指向一个称为 j 的节点/叶节点。在节点 i 和 j 之间，于 T' 中引入一个称为 k 的新节点，并添加边 $i \to k$ 和 $k \to j$。将节点 i 处的标签从 f_i 改为常值函数 1，$i \to k$ 边标记为 1，而 k 标记为函数 f_i。如果 f_i 是单调的，则 $k \to j$ 边标记为 1；否则，$k \to j$ 边标记为 0。根据这一标记原则，注意对于任意输入 x，边 $i \to k$ 始终处于活跃状态；当且仅当 x 在 T 中激活边 e 时，边 $k \to j$ 才处于活跃状态。因此，完成 T' 的构造后，存在一条通向某个叶节点的活跃路径，当且仅当在 T 中存在一条通向同一（对应）叶节点的活跃路径。由于在任一棵树中，通向标签为 1 的叶节点的活跃路径是否存在都是接受判据，因此对于相同的输入，两棵树的输出相同；由此 T' 计算 f。最后，我们

注意， T' 的高度是 T 高度的两倍，因为按照我们的设计， T 中的每条边都会在 T' 中产生两条边。 □

6.2 通过交替数进行刻画

与常规的单调决策树类似，我们在以下定理中证明非确定性单调决策树的高度和规模的界，将它们作为复杂度度量（高度记为 $\mathsf{DT}^n_m(f)$）。

定理 6.1. 对任意布尔函数 f，$\mathsf{DT}^n_m(f) \leq 2$，且最优非确定性单调决策树的规模为 $\lceil \frac{\mathsf{alt}(f)}{2} \rceil$。

证明. 我们知道，任意布尔函数 f 都有一个长度至多为k的单调分解：$= \mathsf{alt}(f) + 1$（引理 3.1），因此令 $f = f_1 \oplus f_2 \oplus \cdots \oplus f_k$。根据 f_i 的蕴含性质，对于 f，我们有以下等价的求值方式，即 $f = \overline{f_1} f_2 \vee \overline{f_3} f_4 \vee \ldots \overline{f_{k-1}} f_k$ [10]（命题 3.1）。现在我们构造一棵高度为 2 的非确定性单调决策树 T，其规模等于上述 f 表示中的“项”数，即 $s := \lceil k/2 \rceil$。 T的根节点包含 s 条边，边的标签分别为 $\overline{f_1}$、$\overline{f_3}$、……、$\overline{f_{k-1}}$，每个标签对应一条边。接着，对根节点的每个子节点，我们添加一个子节点，其边标签为 f_{i+1}，对应于 $\overline{f_i}$。最后，在树的所有叶节点上都附加一个标记为 1 的叶节点。当且仅当 f 在该输入上的求值为 1 时，才存在一条从根节点到叶节点的活跃路径，由此可知 T 计算了 f。

□

7 随机化单调决策树

我们还研究随机化单调决策树。在此模型中，除了决策树中的单调查询节点外，也允许在树的内部节点进行随机比特选择；每个随机选择节点也有两条指向子节点的出边，一条标记为 0，另一条标记为 1。我们称该树计算布尔函数 f，如果对于任意输入 x，计算到达标记为 $f(x)$ 的叶节点的概率（关于树中随机比特选择的取值）至少为 $2/3$。我们用 $\mathsf{DT}^r_m(f)$ 表示计算布尔函数 f 的随机单调决策树的最小高度。以下定理表明，当单调查询不受限制时，随机化并无帮助。

定理 7.1. 对于任意布尔函数 f，$\mathsf{DT}^r_m(f) = \mathsf{DT}_m(f) = \lceil \log(\mathsf{alt}(f) + 1) \rceil$。

证明。 由于任何单调决策树显然也是随机单调决策树，因此我们直接得到 $\mathsf{DT}^r_m(f) \leq \mathsf{DT}_m(f)$。为证明反向不等式，我们将构造一个计算 f的电路，其至多包含 $\mathsf{DT}^r_m(f) - 1$个取反门。然后利用定理 4.3，可得到一棵高度为 $\mathsf{DT}^r_m(f)$ 的（确定性）单调决策树来计算 f。

[10]如果 k 是奇数，我们只需将 f_1 设为常值函数 0。

设 T 是一棵高度为 $h = \mathrm{DT}^r_m(f)$、用于计算 f 的最小高度随机单调决策树。设其中标记为 1 和 0 的叶节点数量分别为 k 和 k'。我们有 $k + k' =$（T 中的叶节点总数）$\leq 2^h$。我们仅在 $k \leq k'$ 时给出证明。否则，可以将 T 的所有叶节点标签取反，从而得到一棵计算 $\overline{f}$ 的随机单调决策树。由于 $\mathrm{DT}_m(f) = \mathrm{DT}_m(\overline{f})$，且 $\mathrm{DT}^r_m(f) = \mathrm{DT}^r_m(\overline{f})$[11],，所需结果即得。因此，我们可以假设 $k \leq k'$。

我们首先为函数 f 推导一个刻画，这也有助于其他证明。

考虑一个固定的输入 x 到 T。根据 RMDT 的定义，$f(x) = 1$ 当且仅当 T 在 x 上的计算到达标记为 1 的叶节点的概率至少为一半 [12]。现在让我们根据 T 的结构来表示这一概率。

设 $\ell_1, \ell_2, \ldots \ell_k$ 为所有标记为 1 的叶节点，$r_1, r_2, \ldots r_k$ 为从根节点到相应叶节点经过的随机节点数；并设 $c_i := (\bigwedge f_{pi}) \wedge (\bigwedge \overline{f_{qi}})$ 表示与 ℓ_i 对应的特征函数，其中 f_{pi} 表示需要通过的单调查询，f_{qi} 表示在从根节点到 ℓ_i 的路径上需要未通过的单调查询。首先，我们计算计算过程中到达特定叶节点 ℓ_i 的概率。得到该值后，由于在任何给定情况下，计算都会到达唯一一个叶节点，因此对于不同 i 而言，上述事件互斥，这意味着所求概率就是计算到达某个特定 ℓ_i 的概率之和。

现在来看到达特定 ℓ_i 的概率：当 $c_i(x) = 0$ 时，该概率为零。事实上，当 $c_i(x) = 0$ 时，根据其定义，我们可以看出，这意味着至少有一个本应通过的单调函数未能通过，或者至少有一个本应未通过的函数却通过了，这两种情况都会导致矛盾。现在，对于 $c_i(x) = 1$ 的情况，概率完全取决于 r_i：由于所有中间的单调查询都支持计算到达 ℓ_i（即 $c_i(x) = 1$），对于路径上的任意随机节点，恰好只有一种结果能使计算沿正确路径到达 ℓ_i。因此，此时的概率等于 $(1/2)^{r_i}$。这两种情况下的概率可以合并写为 $(1/2)^{r_i} c_i(x)$。可以看出，当 $c_i(x) = 0$ 时，概率为 0；当其等于 1 时，概率为 $(1/2)^{r_i}$。

回到总体概率，它等于总和 $\sum_i (1/2)^{r_i} c_i(x)$。因此，我们得到如下刻画：

$$f(x) = 1 \quad \text{iff} \quad \sum_{i=1}^{k} 2^{h-r_i} c_i(x) \geq 2^{h-1}. \tag{1}$$

注意，计算任意 c_i 至多需要一次取反，因为 $c_i = (\bigwedge f_{pi}) \wedge (\neg(\bigvee f_{qi}))$。现在，为了构造一个具有开头所述 $h-1$ 次取反的 f 电路，我们注意到：由于阈值函数是单调的，如果我们能用一个至多进行 $h-1$ 次取反的（多输出）电路得到所有 c_i，就大功告成了。为此，我们采用 Fischer 构造 [11]，其中设计了一个使用 $\lceil \log(m+1) \rceil$ 次取反的多输出电路来计算反相

[11]翻转单调决策树或随机单调决策树中的叶节点标签后，它计算的是补函数。[12]尽管这意味着可以得到更强的 2/3 界，但 1/2 界更便于处理。

函数 $I(z_1, z_2, \ldots, z_m) = (\overline{z_1}, \overline{z_2} \ldots, \overline{z_m})$。取 $m = k$ 和 $z_i = \bigvee f_{qi}$；电路中使用的取反次数为 neg：$= \lceil \log(m+1) \rceil = \lceil \log(k+1) \rceil$。

如果 $k < k'$，那么 neg $= \lceil \log(k+1) \rceil \leq \lceil \log(2^{h-1} - 1 + 1) \rceil = h - 1$。

否则，$k = k' = 2^{h-1}$。假设 T 中最右叶节点的标签为 1。注意，c_k 中没有否定，因此我们无需对 z_k 取反；这样一来，Fischer 构造中使用的输入数量仅为 $k - 1 = 2^{h-1} - 1$。因此，neg $= \lceil \log(2^{h-1} - 1 + 1) \rceil = h - 1$。现在，若 T 中最右叶节点的标签为 0，则翻转 T 中所有叶节点标签，得到一个针对 $\overline{f}$ 的随机单调决策树（RMDT），随后便可归入上述情形（最右叶节点标签为 1）。

根据定理 4.3，现在可以利用这个电路，得到一棵高度为 $h-1+1 = h = \mathsf{DT}^r_m(f)$ 的单调决策树，计算 f。因此，$\mathsf{DT}^r_m(f) = \mathsf{DT}_m(f)$。 □

随机单调决策树模型的一种变体：我们还研究随机模型的一种更强大的变体，其中允许每个节点关联一个由 w 个单调函数构成的多重集（我们称之为查询集）；对于输入 x，在决策树的每个节点，都会从相应的查询集中均匀随机地选择一个查询函数。同样，如果对于任意输入 x，计算到达标签为 $f(x)$ 的叶节点的概率至少为 $2/3$，我们就称这棵树计算布尔函数 f。我们用 $\mathsf{DT}^{R,}{}_m{}^w(f)$ 表示计算 f 的此类随机决策树的最小高度。可以观察到，任何 RMDT 也都能在此模型中实现（查询集的规模为 w）：单调查询f_i可以替换为查询集 $\{f_i, \ldots, f_i$（重复w次）$\}$，而带有随机比特选择的节点则可以替换为查询集 $\{\mathbf{0}, \ldots, \mathbf{0}$（重复$w/2$次）$\mathbf{1}, \ldots, \mathbf{1}$（重复$w/2$次）$\}$。由此，对于任意偶数 $w \geq 2$，可得 $\mathsf{DT}^{R,}{}_m{}^w(f) \leq \mathsf{DT}^r_m(f)$。即使 w 为奇数，该关系仍在相差一个常数因子的意义下成立。对于反方向，我们证明如下结论：

定理7.2. 对任意布尔函数 f，$\mathsf{DT}^r_m(f) \leq (1+k)$。$\mathsf{DT}^{R,}{}_m{}^w(f)$，若 $w = 2^k$ 为整数k。

证明。令 T为对应于更强模型的树，其中查询集的规模为 $w = 2^k$，且其计算 f所需的高度最小。要将 T变换为 RMDT，我们从底向上，在每个内部节点执行以下变换。

设 u 为 T 的一个内部节点，其左分支通向子树 u_L，右分支通向 u_R，且 u处可能的查询函数属于 $\{q_1, q_2, \ldots q_w\}$。考虑一棵完全二叉树B，前 k 层的节点对 0 或 1 进行随机查询，而最底层的 $2^k = w$ 个节点则对相应的 q_i进行查询。

我们用以下树替换以 u 为根节点的 T 子树：在顶部引入 B（替代 u），并将 B 的每个最底层节点分支为左侧子树u_L 和右侧子树 u_R 的副本。显然，对 T的所有内部节点自底向上应用此过程后，我们便得到一个 RMDT。

要证明该变换仍然计算 f，只需说明，对于输入x，由于对应于节点 u的变换，计算采用 u_L 和 u_R 的概率

在变换前后保持不变。事实确实如此，因为在变换前后，这些量分别都等于 $|\{q_i \mid q_i(x) = 0\}|/w$ 和 $|\{q_i \mid q_i(x) = 1\}|/w$ 。

由于每个原始节点都被替换为一棵完全二叉树 B，高度会增加 B （包括最底层）的高度倍，即 $1 + k$。因此，$\mathrm{DT}^r_m(f) \leq (1 + k)$。$\mathrm{DT}^{R,\ w}_{\ m}(f)$。

□

8 带查询限制的单调决策树

本节中，我们研究在单调查询函数受限的情况下，单调决策树的能力。我们将 DT(mon-$\mathcal{C}$) 定义为这样一组布尔函数（更确切地说，是布尔函数族）：它们存在高度为 $\mathcal{O}(\log n)$ 的决策树，其中 n 是变量的数量，且所有查询函数均来自 mon-$\mathcal{C}$。我们也可以考虑多项式规模的决策树，但对于有趣的类 $\mathcal{C}$，这两种表述最终是等价的。类似地，我们将 DL(mon-$\mathcal{C}$) 定义为由多项式规模 MDL 计算、且查询属于 mon-$\mathcal{C}$ 的函数集合。

我们首先说明为何将高度上界设为 $\mathcal{O}(\log n)$。我们证明，对于任意 $h = \omega(\log n)$，都存在一个定义在 n 个变量上的函数 f，它有一棵高度为 h 的决策树，其查询函数可由单调多项式规模电路计算，但 f 无法由多项式规模电路计算。相比之下，对于任意 $h = o(\log n)$，若定义在 n 个变量上的函数 $f \in \mathcal{C}$ 的交替数为 $\Omega(n)$，那么 f 就不存在一棵高度为 h 的决策树，使其查询函数可由 $\mathcal{C}$ 中的单调电路计算。因此，只有当高度为 $\mathcal{O}(\log n)$ 时，DT(mon-$\mathcal{C}$) 是否等于 $\mathcal{C}$ 这一问题才有充分的研究意义。基于这一背景，我们接下来将研究上述定义的 DT(mon-$\mathcal{C}$)。

8.1 DT(mon-$\mathcal{C}$) 的高度约束

命题 8.1. 对任意 $h = \omega(\log n)$，存在一个定义在 n 个变量上的布尔函数 f，它有一棵高度为 h 的决策树，其查询函数可由单调多项式规模电路计算，但 f 无法由多项式规模电路计算。

证明。 我们实际上将证明，存在一个具有“简单决策树”（所有查询都针对输入变量）、高度为 $h = \omega(\log n)$ 的函数，但它不能由任何多项式规模电路计算。根据香农计数论证（见 [15]），我们知道存在一个定义在 h 个变量上的函数，计算它需要规模为 $\Omega(2^h/h)$ 的电路。我们的函数 f 将是同一个函数，只是额外添加了 $(n - h)$ 个虚变量。由于 f 只依赖于 h 个变量，我们可以构造一棵（非自适应）决策树，其查询对象就是该函数所依赖的比特。显然，它的高度为 h。根据定义，任何计算 f 的电路，其规模都为 $\Omega(2^h/h) \geq c_1.(2^h/h)$。 [13]

为了导出矛盾，假设 f 确实有一个多项式规模的电路。这意味着$c_1.(2^h/h) \leq n^{c_2}$。取对数后，我们得到 $c_3 + h - \log\ h \leq c_2 \log\ n$，因此有， $h \leq 2h - 2\log\ h \leq$

[13] c_1、c_2、c_3 是一些固定的正数常量，并且这些不等式是渐近的，即对于足够大的 n。

2.($c_3 + h - \log\ h$) $\leq 2c_2 \log n = \mathcal{O}(\log\ n)$，这与 $h = \omega(\log\ n)$ 矛盾。因此，f无法由多项式规模电路族计算。 □

命题 8.2. 设 $\mathcal{C}$ 是一个包含函数 f 的电路复杂度类，且 alt(f) =Ω(n)。对于任意 $h = o(\log n)$，存在一个定义在 n 个变量上的函数 $f \in \mathcal{C}$，它不具有高度为 h的决策树，且其查询函数可由 $\mathcal{C}$中的单调电路计算。

证明。 设 $f \in \mathcal{C}$ 是一个满足以下条件的函数：$\mathsf{alt}(f) = \Omega(n)$。反设$f$确实具有一棵高度为 $h = o(\log\ n)$的单调决策树。我们知道 $\mathsf{alt}(f) \leq 2^h$，其中 h 是决策树的高度（回忆定理 4.2）。于是，alt(f) $\leq 2^{o(\log n)} = o(n)$，这与 alt($f$) $= \Omega(n)$ 矛盾。

□

8.2 查询限制下的确定性MDT：DT(mon-$\mathcal{C}$) 与 $\mathcal{C}$

如引言中所述，我们要问：查询函数可由类 $\mathcal{C}$中的单调电路计算的单调决策树计算，能够在多大程度上模拟类 $\mathcal{C}$中的一般计算？沿着这个方向，我们首先证明，当 $\mathcal{C}$ 具有合理的闭包性质时，DT(mon-$\mathcal{C}$) $\subseteq \mathcal{C}$。

引理 8.1. 对于一个在多项式多个 $\neg$、$\wedge$、$\vee$ 操作下封闭的电路复杂度类 $\mathcal{C}$，DT(mon-$\mathcal{C}$) $\subseteq \mathcal{C}$。

证明。定理 4.1 的证明也表明，DT(mon-$\mathcal{C}$) = DL(mon-$\mathcal{C}$)。由于对数高度单调决策树与多项式规模单调决策列表可以相互转换，因此只需证明 DL(mon-$\mathcal{C}$) $\subseteq \mathcal{C}$。设布尔函数 f 通过决策列表 $L = (f_1, c_1)(f_2, c_2) \ldots (f_k, c_k)$ 属于 DL(mon-$\mathcal{C}$)，其中k =poly(n)；且每个查询函数 f_i 都有一个来自类 $\mathcal{C}$ 的（单调）电路 C_i。利用具有上述性质的电路类对应的决策列表标准形式，我们可以假设各个 c_i 交替出现；并且查询函数 f_i 是前向触发的，即 $f_1 \Rightarrow f_2 \Rightarrow \cdots \Rightarrow f_k$。由于我们可以在开头添加一个 (0, 0) 节点，或在结尾添加一个 (1, 1) 节点，同时仍保持标准形式，因此不失一般性，我们可以假设 $c_1 = 0$ 且 $c_k = 1$（因此 k 为偶数）。根据交替常数性质，这意味着 $c_{2i} = 1$ 且 $c_{2i-1} = 0$。

根据命题 3.1 我们知道，布尔函数 $g := \overline{f_1} f_2 \vee \overline{f_3} f_4 \cdots \vee \overline{f_{k-1}} f_k$ 等价于 f。最后，我们注意到，由于 $\mathcal{C}$ 和 $k = \mathsf{poly}(n)$ 具有闭包性质，表达式g 可用于构造一个 $\mathcal{C}$ 电路来计算 f。

□

如果该类 $\mathcal{C}$ 足够丰富，能够包含阈值函数的单调电路，例如类 TC^0 本身，那么我们实际上可以证明二者相等：注意，引理 3.2 给出的单调分解，可以利用命题 3.1轻松转换为一个 MDL，其中查询就是相同的函数。因此，我们得到 $\mathcal{C} \subseteq$ DL(单调-$\mathcal{C}$)，结合以下事实：DT(单调-$\mathcal{C}$) =DL(单调-$\mathcal{C}$)，以及引理 8.1 即可完成定理 1.2的证明。

定理 **1.2**。 对于任意电路复杂度类 $\mathcal{C}$，只要单调-$\mathsf{TC}^0 \subseteq$单调-$\mathcal{C}$，就有 DT(单调-$\mathcal{C}$) $= \mathcal{C}$。

8.3 单调决策树与 AC^0

我们尝试回答这个问题：DT(单调-AC^0) 与 AC^0。我们知道，根据引理 DT(单调-AC^0) =DL(单调-AC^0) 包含于 AC^0，引理 8.1。一个有趣的挑战是证明反向包含关系是否成立。作为热身，我们将证明，DT(单调-AC^0) 比多项式规模的项决策列表（它们是 AC^0 的真子集）更强大。

命题 8.3. DT(单调-AC^0) $\not\subseteq$项决策列表。

证明。我们证明 DL(单调-AC^0) $\not\subseteq$项决策列表。根据引理 8.1 中的构造，我们知道，项决策列表中的所有函数都有多项式规模、深度为 4 的电路，因为查询函数是深度为 2 的单调电路。因此，如果我们能证明存在一个函数 f，它属于 DL(单调-AC^0)，但没有多项式规模、深度为 4 的电路，就能得出结论。众所周知，对于任意深度 d，都存在一个单调函数 f，它可以由深度为 d 的单调电路计算，但无法由任何多项式规模、深度为 $d-1$ 的电路（甚至是非单调电路）计算（见 [20]）。由于决策列表 $L = (f, 1)(1, 0)$ 可以计算 f，且查询函数 f 有一个单调 AC^0 电路，因此 $f \in$ DL(单调-AC^0)。而且 $f \notin$ 不属于项决策列表，因为项决策列表中的所有函数都有深度为 4 的 AC^0 电路，而根据定义，f 没有这样的电路。

□

为了进一步将该类与 AC^0, 进行比较，我们首先将命题 8.2[14] 和 8.1 应用于 AC^0，并得出：对于任意 $g(n) = o(\log n)$，以及 $h(n) = \omega(\log n)$，$\mathsf{DT}^{g(n)}$(mon-AC^0) $\subsetneq \mathsf{AC}^0$，且 $\mathsf{DT}^{h(n)}$(mon-AC^0) $\not\subseteq \mathsf{AC}^0$。与此相反，我们证明整个 AC^0 都可以由具有某种次线性高度的单调决策树计算。通过使用 Santha 和 Wilson 的一个定理（见 [18]中的定理 4.1），该定理将电路中的取反数量减少到 $^n\log^r n$，然后应用定理 4.3，我们证明：

定理 8.1。对于任意常数 r，$\mathsf{AC}^0 \subseteq \mathsf{DT}^{d(n)}$(mon-$\mathsf{AC}^0$)，其中 $d(n) = \Omega\left(^n\log^r n\right)$。

证明。我们使用 Santha 和 Wilson 提出的以下定理（参见 [18]中的定理 4.1）：对于任意常数 r，每个AC^0 电路族都可以转换为一个常数深度、多项式规模的电路，且最多使用 $\mathcal{O}\left(^n\log^r n\right)$ 个取反。我们可以直接将其与定理4.3结合，得到一棵高度至多为 $\mathcal{O}\left(^n\log^r n\right)$ 的决策树。注意，由于定理4.3 证明中使用的查询是原电路的单调子电路[15]，在这种情况下，它们可以由单调AC^0计算。

□

[14]应用命题 8.2时，取 $f(x) = x_1\overline{x_2} \vee x_3\overline{x_4} \cdots \vee x_{n-1}\overline{x_n}$。 [15]子电路是指通过将某些中间门的值固定为 0 或 1 而得到的电路。

DT(mon-AC^0) 的取反受限计算：我们证明，DT(mon-AC^0) 中的函数具有取反门数量“受限”的 AC^0 电路。

定理**1.3.** 若布尔函数 f 在 n 个变量上属于DT(单调-AC^0)，那么对于任意正数常数 $\epsilon \leq 1$，都存在一个AC^0 电路，用于计算 f，且包含 $\mathcal{O}(n^\epsilon)$个取反门。

证明。由于 $f \in$DL(单调-AC^0)，假设 MDL 处于标准形式，并应用命题 3.1，我们可以写出 $f = \overline{f_1} f_2 \vee \overline{f_3} f_4 \vee \ldots \overline{f_{\ell-1}} f_\ell$，其中 $\ell = \mathcal{O}(n^k)$，某个常数为 k。此外，所有 f_i 都有单调 AC^0 电路，且 $\forall i \in [\ell - 1], f_i \Rightarrow f_{i+1}$。因此，只需利用一个使用 $\mathcal{O}(n^\epsilon)$ 个取反的常数深度多项式规模电路，从 $f_1, \ldots f_n$ 为每个为奇数的 $i \in [\ell]$ 生成 $\overline{f_i}$。事实上，平凡电路使用 $\ell = \mathcal{O}(n^k)$ 个取反。

主要的观察是，我们需要取反的比特（ f_i where i 为奇数时的输出）已经按排序顺序排列，因为 $\forall i, f_i \Rightarrow f_{i+1}$。将这个比特串记为 $s = 0^j 1^{m-j}$，其中 $m := \lceil \ell/2 \rceil$。我们需要输出 $\overline{s} = 1^j 0^{m-j}$。

正如 [18]（定理3.6，见 [18]）中所证明的，这可以通过一种迭代构造来实现，该构造仅使用 $\mathcal{O}(n^\epsilon)$ 次取反。在 [18]，中定理 3.6 的证明里，作者还指出，他们的构造这一部分仅使用多项式数量的 $\neg$，以及（无界的） $\wedge$ 与 $\vee$ 门。因此，最终电路属于 AC^0。我们在附录A.2中结合本文背景给出该构造，以供参考。

□

相较之下，我们注意到，某些 AC^0 电路需要大量取反。

定理 8.2（[18] 的定理3.2）。对于每个满足 alt(f) $= \Omega(n)$ 的 $f \in$ AC^0，以及每个 $\epsilon > 0$，任何计算 f 的 AC^0 电路，对某个正数常量 ϵ（可能取决于该电路）而言，至少需要 $\Omega(n^\epsilon)$ 个取反门。

证明。由 [18] 可直接得出这一结论，作者在其中证明：[16], 如果某个交替数为 k 的布尔函数 f 由一个深度为 d 的电路 C 计算，那么 C 中的取反数至少为$d(k+1)^{1/d} - d$。由于 d 是常数且 $k = \Omega(n)$，这就是 $\Omega(n^\epsilon)$。 □

因此，如果定理 8.2 针对某个 $f \in$AC^0 和固定的 ϵ 可以在渐近意义上得到改进，那么我们就能证明 DT(单调-AC^0) 严格包含于 AC^0。

DT(mon-AC^0) 与 AC^0 的分离候选函数问题：现在我们证明，存在一个简单函数，可以由深度为二的 AC^0 电路计算；如果能证明它属于 DT(mon-AC^0)，就能推出 DT(mon-AC^0) $=$ AC^0。这也为分离这两个类提供了一个潜在的候选函数。

定理 8.3. 若函数族 $f^n(x) = \overline{x_1} x_2 \vee \overline{x_3} x_4 \vee \ldots \overline{x_{n-1}} x_n$ 属于 DT(mon-AC^0)，则 DT(mon-AC^0) $=$ AC^0。

[16]尽管这一结论是针对 [18], 中的多输出函数提出的，但它对单输出函数同样成立。

证明。假设函数 f^n 属于 $\mathsf{DT}(mon\text{-}\mathsf{AC}^0) = \mathsf{DL}(mon\text{-}\mathsf{AC}^0)$。假设 MDL 处于标准形式，根据命题 3.1，$f^n$ 可以表示为以下几种形式：

$$\begin{aligned} f^n &= (f_1^n, 0)(f_2^n, 1)\dots(f_{p(n)}^n, 1)(f_{p(n)+1}^n, 0) \\ &= f_1^n \oplus \cdots \oplus f_{p(n)+1}^n \\ &= \overline{f_1^n} f_2^n \vee \overline{f_3^n} f_4^n \vee \cdots \vee \overline{f_{p(n)-1}^n} f_{p(n)}^n \\ &= (f_2^n \vee \overline{f_3^n}) \wedge (f_4^n \vee \overline{f_5^n}) \wedge \cdots \wedge (f_{p(n)}^n \vee \overline{f_{p(n)+1}^n}), \end{aligned} \tag{2}$$

其中，对于所有 i，$f_i^n \in$mon-AC^0，和 $f_i^n \Rightarrow f_{i+1}^n$，且 $p(n)$ 为偶数，并且是 n 的某个多项式。假设 $p(n) = n^{c_1}$，且每个 f_i^n 都有一个深度为 d、规模至多为 n^{c_1} 的单调 AC^0 电路，其中 $d \geq 1, c_1 \geq 2$ 为某些常数。

现在我们将证明 $\mathsf{AC}^0 \subseteq$ DT(mon-AC^0)。考虑任意一个由深度至多为 e、规模至多为 n^{c_2} 的德摩根公式[17] G 计算的函数 $g \in \mathsf{AC}^0$，其中 e 和 $c_2 \geq 2$ 为某些常数。我们将证明，g 有一个 MDL L_g，其（标准形式下的）规模至多为 $n^{(c_1c_2)^{2e}}$，且所有查询的电路深度至多为 de、规模至多为 $n^{(c_1c_2)^{2e}}$。那么，由于 e, c_1, c_2 都是常数，单调电路的规模和查询数量都是 n 的多项式，因此 $g \in$DL(mon-AC^0)。

我们通过对 e 进行归纳法，证明 L_g 的存在。

基础情形：$e = 1$。深度为 1 的公式就是文字的或或与。若 $g = (\bigvee_i x_i) \vee (\bigvee_j \overline{x_j})$，其中 x_i 和 x_j 都是变量，那么 $L_g = (\bigvee_i x_i, 1)(\bigwedge_j x_j, 0)(1, 1)$ 即可。或门的情形也可类似处理。因此，L_g 的深度、规模和查询次数都满足预期的界限（假设 n 不太小）。

归纳步骤：$e \geq 2$。同样，我们假设根门是一个或门：$g = \bigvee_{i=1}^s h_i$，其中每个 h_i 都有一个深度至多为 $e-1$、规模至多为 n^{c_2} 的 AC^0 公式。根据归纳假设，这意味着它有一个规模至多为 $n^{(c_1c_2}$ 的 MDL $L_{h_i})^{2e-2}$，使得其所有查询都有深度至多为 $d(e-1)$、规模至多为 t 的单调电路：$= n^{(c_1c_2)^{2e-2}}$ ——设 $L_{h_i} = f_{i,1}f_{i,2} \vee f_{i,3}f_{i,4} \vee \cdots \vee f_{i,t-1}f_{i,t}$。于是，我们有 $g = \bigvee_{i=1}^s h_i = \bigvee_{i=1}^s ($ $f_{i,1}f_{i,2} \vee f_{i,3}f_{i,4} \vee \cdots \vee f_{i,t-1}\overline{f_{i,t}})$。现在的诀窍是注意到，这个关于 g 的表达式看起来恰好像布尔函数 f^n，只是其中的变量被一些单调函数 $f_{i,j}$ 替换了，而它们各自都有深度至多为 $d(e-1)$、规模至多为 t 的单调电路。也就是说，$g = f^{st}(f_{1,1}, \dots, f_{1,t}, f_{2,1}, \dots, f_{2,t}, \dots, f_{s,1}, \dots, f_{s,t})$。现在，利用我们根据等式（2）得到的关于$f$ 的 MDL（族），通过将变量 x_i 替换为函数$f_{i,j}$，我们得到关于 g 的 MDL L_g。查询数量为 $p(st) = (st)^{c_1}$，每个查询都有一个深度至多为 $d + d(e-1) = de$、规模至多为 $(st)^{c_1} + st.t \leq (st)^{c_1+2} \leq (n^{c_2}n^{(c_1c_2)^{2e-2}})^{c_1+2} \leq n^{(c_2+(c_1c_2)^{2e-2})(c_1+2)} \leq n^{(c_1c_2)^{2e}}$ 的单调公式（利用 $e, c_1, c_2 \geq 2$）。

[17] AC^0 等价于多项式规模的常数深度德摩根公式，即取反仅作用于输入变量的布尔公式。不失一般性，我们还假设公式中与门和或门位于交替的层级。

如果根门是与门，类似的构造也适用；此时我们可以使用 f 的“合取范式”，而不是“析取范式”。

□

8.4 带查询限制的随机化MDT

与确定性情形类似，当高度限制为 $\mathcal{O}(\log n)$ 时，我们可以为电路复杂度类 $\mathcal{C}$ 定义 RDT(mon-$\mathcal{C}$)。通过使用阈值门计算概率界，我们证明 RDT(mon-$\mathcal{C}$) $= \mathcal{C}$ if mon-TC$^0 \subseteq$ mon-$\mathcal{C}$。随后，通过为随机单调决策树精心构造一种标准形式，我们证明 RDT(mon-AC^0) $\subseteq$AC^0。

定理 8.4. 对于任意满足 mon-TC$^0 \subseteq$mon-$\mathcal{C}$ 、且在多项式 $\vee$ 下封闭的电路复杂度类 $\mathcal{C}$ ， $\wedge, \neg$，我们有 RDT(mon-$\mathcal{C}$) $= \mathcal{C}$。

证明。为证明 RDT(mon-$\mathcal{C}$) $\subseteq \mathcal{C}$，令 $f: \{0,1\}^n \to \{0,1\}$ 为 RDT(mon-$\mathcal{C}$) 中的任意函数。根据定义，存在一个称为 T 的随机单调决策树，其高度为 $h = \mathcal{O}(\log\ n)$。令 $\ell_1, \ell_2, \ldots \ell_k$ 为所有标记为 1 的叶节点； $r_1, r_2, \ldots r_k$ 为从根节点到相应叶节点的随机节点数；并令 $c_i := \bigwedge f_{pi} \bigwedge \overline{f_{qi}}$ 表示与 ℓ_i 对应的特征函数，其中 f_{pi} 表示根节点到叶节点路径上需要通过的单调查询， f_{qi} 表示需要未通过的查询。注意，由于这些查询具有属于 $\mathcal{C}$ 的电路，因此 c_i 函数也如此。根据定理 7.1 的证明中的刻画（公式 (1)），我们有 $f(x) = 1$ 当且仅当 $\sum_i 2^{h-r_i} c_i(x) \geq 2^{h-1}$。

右侧可以写成 $\sum_i P_i(n) c_i(x) \geq Q(n)$，其中系数 P_i 和 Q 是 n 的多项式，因为 $h = \mathcal{O}(\log n)$。

我们有 $f(x) = 1 \iff \sum_i P_i(n) c_i(x) \geq Q(n)$。由于多项式加权阈值可以在 TC^0 中实现（因此在 $\mathcal{C}$ 中有一个电路）， c_i 也同样如此，我们可以为右侧的任务构造一个属于类 $\mathcal{C}$ 的电路，而这显然就是左侧的函数 f。

因此，我们有 RDT(mon-$\mathcal{C}$) $\subseteq \mathcal{C}$。由于 DT(mon-$\mathcal{C}$) $\subseteq$RDT(mon-$\mathcal{C}$)，并且我们已经证明 DT(mon-$\mathcal{C}$) $= \mathcal{C}$ （定理 1.2），所以得到 RDT(mon-$\mathcal{C}$) $=$ DT(mon-$\mathcal{C}$) $= \mathcal{C}$。 □

对于 AC^0，我们关于查询受限的随机化单调决策树有一个部分结果 :

定理 8.5。RDT(mon-AC^0) $\subseteq$ AC^0。

在描述证明之前，我们先为随机单调决策树模型推导标准形式。设任意布尔函数 f 在 n 个变量上由一个称为 T 的 RMDT 计算。我们将把 T 修改为一个 RMDT T'，使得 T' 也计算 f，并且 T' 还满足以下性质。

- 从 T' 的根节点到叶节点的所有路径上，随机节点的数量都相同。为此，设 T中任意一条从根节点到叶节点的路径上随机节点数量的最大值为 r ，并设 T 中有一个叶节点 ℓ ，其到根节点的路径上包含 d个（$< r$）随机

节点。然后，我们将叶节点 ℓ 替换为一个随机节点，并将它的两个子节点都设为叶节点，且标签与 ℓ相同。现在可以看出，新添加的两个叶节点到 T根节点的距离都是$d+1$。需要注意的是，这项修改不会改变到达正确叶节点的概率。如果在一次计算中，以某个概率到达原叶节点 ℓ，那么以相同的概率就会到达 ℓ 的某个子节点（具体到达哪一个无关紧要，因为它们的标签都与 ℓ的标签相同）。我们对 T 中所有到根节点 T的距离小于 h 的叶节点执行此操作，直到不存在这样的叶节点。因此，在最终的树 T'中，所有路径都包含相同数量的随机节点（即r个）。

- T' 是一棵完全二叉树。如果 T 已经满足此性质，我们就完成了。否则，假设 ℓ 是 T 的某个叶节点，它到根节点的距离（即查询次数）为 $d < h$，其中 h 是 T的高度。现在考虑对 T进行如下变换。我们用单调查询 1 替换 ℓ，并任意标记其左侧叶节点，再用 ℓ的标签标记右侧叶节点。可以看出，新增的两个叶节点到 T的根节点的距离为 $d+1$。此外， T 仍然计算相同的函数：无论何时，在原始 T中对某个输入进行计算时到达叶节点 ℓ，新的 T 中的计算都会在新增的右侧叶节点处结束，而该叶节点的标签与 ℓ相同。我们对所有到根节点距离小于 h 的叶节点重复执行此变换，直到所有叶节点到根节点的距离都为 h。

注意，通过对 T' 执行相同的变换，可以使其同时满足上述两个性质；但应先使所有路径中的随机节点数量相同，然后再使所有路径长度相同。

- 所有随机节点都位于 T'的顶层。也就是说，在从根节点到叶节点的任意路径中，前几个节点是随机节点，其余节点是单调节点。为此，我们首先对 T执行上述两种变换，之后可以假设每条路径中都有 r个随机节点，且每条路径的长度都等于 T的高度 h。

实现这一点的思路是：我们不通过单调查询在计算过程中进行随机查询，而是先固定一种“随机性”，然后再通过单调查询继续计算。

我们将 T的许多确定性版本定义为集合 $\{T_s \mid s$，其中每个元素都是长度为 $r\}$的二进制序列。树 T_s 与 T 在单调节点、叶节点及其标签方面具有相同的结构。若 s 的第 i^{th} 个比特为 0，则将随机节点 v 替换为单调函数 0，否则替换为单调函数 1；其中， i 是从 v 到 T的根节点路径上的随机节点数。

新树 T' 的高度将为 $r+h$，其构造如下： T'的顶部应为一棵高度为 r 的完全二叉 MRDT，其中所有节点都进行随机查询。这会产生 2^r 个“悬空端”。对于所有二进制序列 s，由二进制

序列 s 所寻址的悬空端应通向 T_s 的根节点。至此， T' 的构造完成。立即可以看出， T' 满足所有随机节点都位于顶部这一所需性质。现在，为了证明 T' 等价于 T，我们将论证：对于任意输入 x，在 T 中到达 1 的概率等于在 T' 中到达 1 的概率。前者等于 $\sum_i (1/2)^r.c_i(x)$，其中 i 遍历 T 中所有标记为 1 的叶节点 ℓ_1、ℓ_2、.....，而 c_i 是它们各自的特征函数，其定义见定理 17。由于 T' 中对应于 T 的每个叶节点 ℓ_i 都有 2^r 个叶节点，后一个概率等于 $\sum_i \sum_s (1/2)^r.c_i^s(x)$。这里， c_i^s 表示树 T_s 中与 ℓ_i 对应的叶节点的特征函数。为了证明这两个概率相等，我们将证明对于任意 i， $c_i(x) = \sum_s c_i^s(x)$。由于树 T_s 与 T 相同，只是将随机节点改为 0 或 1，因此可以观察到，对于任意 s， $c_i^s(x) \Rightarrow c_i(x)$。现在，假设对于某个 s，$c_i^s(x) = 1$。除 $c_i(x) = 1$ 外，这还意味着，从根节点到 ℓ_i 的 T_s 路径上仅对应于原始随机节点的特征乘积为 1。这意味着乘积中的每个文字都为 1，也就是说，我们可以确定 T_s 的每个原始随机节点是 0 还是 1（因为我们知道从根节点到 ℓ_i 的路径）。但根据 T_s 的定义，这唯一确定了一个 s。因此， $c_i^s(x)$ 中至多有一个会为 1。结合这一点以及 $\forall s$， $c_i^s(x) \Rightarrow c_i(x)$，我们得到所需的方程 $c_i(x) = \sum_s c_i^s(x)$。因此， T' 也计算 f。

定理 *8.5* 的证明。 设布尔函数 f 在 n 个变量上有一个高度为$h = \mathcal{O}(\log n)$ 的随机单调决策树，记作 T，其所有单调查询都具有单调 AC^0 电路。根据上面证明的三种标准形式，我们可以假设 r 层 T 的前几层是随机节点，且其下方的查询都具有单调 AC^0 电路。这样做是合理的，因为这些标准形式不会使树高超过 $\mathcal{O}(\log n)$，且查询函数仍具有单调 AC^0 电路。

由于在高度 r 之后没有随机查询，因此在 T 中，每个紧邻随机节点下方节点为根的子树都类似于一棵（确定性）单调决策树。既然我们已经证明了 $\mathsf{DT(mon\text{-}AC^0)} \subseteq \mathsf{AC}^0$（引理 8.1），那么这些子树各自计算一个 AC^0 函数，例如 f_1、f_2、$\ldots f_{2^r}$。对于输入x，回想一下，这意味着在 T 中，到达标签为 $f(x)$ 的叶节点的概率至少为 2/3；换句话说，所有能以非零概率到达的叶节点中，至少有 $2/3^{rd}$ 的叶节点标记为 $f(x)$。由于子树是确定性的，对于给定的 x，每棵子树中到达的叶节点唯一。因此，对于输入 x，至少有 $2/3^{rd}$ 的子树到达标记为 $f(x)$ 的叶节点（相应地， $f_i(x) = f(x)$）。因此， f 本质上是一个对 f_i 求多数的函数，而且多数位所占比例至少为总数的 $2/3^{rd}$。Ajtai 和 Ben-Or [1]证明了存在一个 AC^0 电路，在这种情况下可以计算多数函数。由于所有 $2^r = \text{poly}(n)$ 个函数 f_i 也都属于 AC^0,，我们可以为 f 构造一个 AC^0 电路。因此， $f \in \mathsf{AC}^0$，且 $\mathsf{RDT(mon\text{-}AC^0)} \subseteq \mathsf{AC}^0$。

□

9 讨论与开放问题

我们探讨确定性单调决策树（自适应和非自适应）以及采用最一般单调查询的 MDL 的能力，并确立相应复杂度度量与交替数之间的关系（第 4 和 5节）。我们还引入 NMDT 和 RMDT，并研究它们的能力（第 6 和 7节）。针对受限查询的情形，我们证明了各种电路复杂度类与相应的确定性和随机化 MDT 类之间的包含关系（第 8节）。利用 [18], 提出的一种构造，我们证明了 AC^0 电路中取反次数的上界，并阐明了它在潜在地解决 $\mathsf{DT}(\text{单调-}\mathsf{AC}^0) =_? \mathsf{AC}^0$中的作用（小节 8.3）。

$\mathsf{DT}(\text{mon-}\mathsf{AC}^0) =_? \mathsf{AC}^0$ 的问题是我们尚未解答的主要问题之一。甚至尚不清楚像 $f = \overline{x_1}x_2 \vee \overline{x_3}x_4 \vee \cdots \vee \overline{x_{n-1}}x_n$ 这样简单的函数是否属于 $\mathsf{DT}(\text{mon-}\mathsf{AC}^0)$，甚至是否属于 $\mathsf{RDT}(\text{mon-}\mathsf{AC}^0)$。在这方面，我们首先指出，定理 1.3 中取反次数的渐近上界无法改进；因为如果能够改进，我们就能证明 TC^0 中的任意函数都可以由一个 TC^0 电路计算，且只需 $o(n^\epsilon)$ 次取反（对任意 $0 < \epsilon < 1$ 均成立），这与 [18] 的推论 3.3(2) 矛盾。我们注意到，如果能够针对某个函数改进定理 8.2 中的取反次数上界，就能得到该分离结果。若无法改进，则意味着 AC^0 中的每个函数都可以用一个 AC^0 电路计算，且只需 $\mathcal{O}(n^\epsilon)$ 个取反门，其中常数 $\epsilon > 0$ 可以任意小；这将改进定理 8.1。

这里我们还讨论另一种定义类 $\mathsf{DT}(\text{mon-}\mathcal{C})$、$\mathsf{RDT}(\text{mon-}\mathcal{C})$ 和 $\mathsf{DL}(\text{mon-}\mathcal{C})$ 的方式：我们可以通过要求查询是单调的来限制查询，但允许其在 $\mathcal{C}$ 中使用可能非单调的电路，而非单调电路。这个版本可能会得到更强大的类。通过非常相似的证明，我们仍可得到 $\mathsf{DT}(\text{mon-}\mathcal{C}) = \mathsf{RDT}(\text{mon-}\mathcal{C}) = \mathcal{C}$when $\mathcal{C} \supseteq \mathsf{TC}^0$。此外，命题 3.3 意味着，对于任意具有一致交替的函数 $f \in \mathsf{TC}^0$，我们有 $f \in \mathsf{DT}^{\lceil \log(\mathsf{alt}(f)+1) \rceil}(\text{mon-}\mathsf{TC}^0)$。这对我们的其他结果有何影响尚不明确。

参考文献

[1] Miklos Ajtai 和 Michael Ben-Or。关于概率常数深度计算的一个定理。载于第十六届 ACM 年度计算理论研讨会论文集，第 471–474 页，1984 年。 31[2] Martin Anthony。决策列表。技术报告，2002 年。CDAM 研究报告 LSE-CDAM-2005-23。 7[3] Y. Ben-Asher 和 I. Newman。使用与、或查询的决策树。载于 第十届复杂性理论结构年度会议论文集。，第 74–81 页，1995 年。 3[4] Michael Ben-Or。代数计算树的下界。载于 第十五届 ACM 计算理论研讨会论文集，第 80–86 页，1983 年。 3

[5] Anders Bj¨orner、L´aszl´o Lov´asz 和 Andrew C. C. Yao。线性决策树：体积估计与拓扑界。载于第 24 届 ACM 计算理论研讨会论文集，第 170–177 页，1992 年。3[6] Eric Blais、Cl´ement L. Canonne、Igor C. Oliveira、Rocco A. Servedio 和 Li-Yang Tan。学习取反数目较少的电路。载于近似、随机化与组合优化，第 40 卷，第 512–527 页，2015 年。8、9、34[7] Avrim Blum。秩-r决策树是 r-决策列表的子类。《信息处理快报》，42(4)：183–185，1992 年。13[8] Nader H. Bshouty。使用等价查询学习 DNF 的亚指数精确学习算法。载于《信息处理快报》，第 37–39 页，1996 年。7[9] Harry Buhrman 和 Ronald de Wolf。复杂度度量与决策树复杂度：综述。《理论计算机科学》，288(1)：21–43，2002 年。3[10] David Dobkin 和 Richard J. Lipton。背包问题线性搜索程序的 n 下界。《计算机与系统科学杂志》，第 413–417 页，1978 年。3[11] Michael J Fischer。否定受限网络的复杂度——简要综述。载于《自动机理论与形式语言》，第二届 GI 凯撒斯劳滕会议，第 71–82 页，1975 年。22[12] Parikshit. Gopalan、Ryan. O'Donnell、Rocco A. Servedio、Amir. Shpilka 和 Karl. Wimmer。傅里叶维数与稀疏性的测试。《SIAM计算杂志》，40(4)：1075–1100，2011 年。3[13] David Guijarro、Victor Lavin 和 Vijay Raghavan。单调项决策列表。《理论计算机科学》，259(1)：549–575，2001 年。7[14] Hao Huang。超立方体的诱导子图与敏感度猜想的证明。《数学年刊》，190(3)：949–955，2019 年。3[15] Stasys Jukna。《布尔函数复杂性：进展与前沿》。Springer Berlin Heidelberg，柏林，海德堡，2012 年。3、6、24[16] Eyal Kushilevitz 和 Yishay Mansour。利用傅里叶谱学习决策树。《SIAM计算杂志》，22(6)：1331–1348，1993 年。3[17] A. A. Markov。关于函数系统的反演复杂度。《ACM杂志》，5(4)：331–334，1958 年 10 月。16[18] M. Santha 和 C. Wilson。限制常深度电路中的取反。《SIAM计算杂志》，22(2)：294–302，1993 年。2、5、26、27、32、35

[19] Swagato Sanyal。傅里叶稀疏度与维数。 计算理论，15(11)：1–13，2019。 3[20] Michael Sipser。博雷尔集与电路复杂度。见第十五届 ACM 计算理论研讨会年会论文集，STOC 83，第61–69页，1983年。 26[21] Marc Snir。证明线性决策树的下界。见自动机、语言与程序设计国际会议，第305–315页，1981年。 3[22] J Michael Steele 和 Andrew C Yao。代数决策树的下界。技术报告，斯坦福大学计算机科学系，1980年。 3[23] Andrew C. Yao 和 Ronald L. Rivest。关于多面体判定问题。 SIAM计算杂志，9(2)：343–347，1980年。 3[24] Zhiqiang Zhang 和 Yaoyun Shi。关于布尔函数的奇偶复杂度度量。《理论计算机科学》，411(26)：2612–2618，2010年。 3

A 附录

A.1 单调分解引理的证明（引理 3.1）

证明。我们重述 [6] 中引理 1 的证明，并补充详细内容，以证明我们所需的额外性质。回顾一下，链 $\mathcal{X}$ 相对于函数 f 的交替次数 alt(f, X)，是沿着 $\mathcal{X}$ 移动时 f 的取值发生变化的次数。

对于输入 $x \in \{0,1\}^n$，我们将 $^{a}f(x)$ 定义为：$= max\{\text{alt}(f, \mathcal{X})\}$，其中 $\mathcal{X}$ 是一条从 x 开始的任意链。注意，alt(f) $= max_x\{a_f(x)\} = a_f(0^n)$。我们对 $k =$alt(f) 使用归纳法来证明该引理。

基础情形 alt(f) $= 0$ 意味着，对于所有输入， f 或为 0，或为 1。当 $f(0^n) = 0$ 时，我们有 $f(x) = 0$；而当 $f(0^n) = 1$ 时，我们有 $f(x) = 1 = \neg(0)$。因此，我们可以轻易地将 f 分解为 $k = 0$ 个单调函数。

现在假设对于所有定义在 n 个变量上且交替数小于 k 的函数，该结论都成立，并证明它对满足 alt(f) $= k$ 的函数 f 也成立。对于 $1 \leq i \leq k$，我们定义如下的 n 变量布尔函数：

$$g_i(x) = 1 \iff a_f(x) < i.$$

我们将证明这些 g_i 实际上就是所需的单调分量 f_i。首先注意，所有 g_i 都是单调的，因为对于任意一对输入 $x \prec y$，都有 $^{a}f(y) \leq a_f(x)$，因此 $g_i(x) \leq g_i(y)$。此外，蕴含性质也成立，因为 $g_i(x) = 1 \Rightarrow a_f(x) < i < i+1 \Rightarrow g_{i+1}(x) = 1$。

我们将论证，当 $f(0^n) = 0$ 时， f确实等于 $g_1 \oplus g_2 \oplus \cdots \oplus g_k$。对于 $f(0^n) = 1$ 的情形，也可以进行类似的推导。考虑函数 $f' := f \vee \overline{g_k}$。我们将证明 f' 的以下性质。

- $f'(0^n) = 1$: 由于 $\mathsf{alt}(f) = k$，至少存在一条链（记作 $\mathcal{X}^*$），其相对于 f 的交替数为 k。因此，$a_f(0^n) = a(f, \mathcal{X}^*) = k$，且 $g_k(0^n) = 0$，这意味着 $f'(0^n) = f(0^n) \vee g_k(0^n) = 1$。

- $\mathsf{alt}(f') = k-1$: 我们将证明，对于任意链 $\mathcal{X}^* \equiv \langle x(1)\, x(2) \ldots x(l) \rangle$，若 $\mathsf{alt}(f, \mathcal{X}^*) = k$，则 $\mathsf{alt}(f', \mathcal{X}^*) = k-1$。注意，由于 $f(0^n) = 0$，存在首个索引 p 使得 $f(x(p)) = 1$（因为 $k > 0$）。根据 g_k 的定义，$x(1), \ldots, x(p-1)$ 中的每一项都不满足 g_k（对于任意 $1 \le i \le p-1$，从 $x(i)$ 开始的 $\mathcal{X}^*$ 的后缀链 $\mathcal{C}_i$ 都有 $\mathsf{alt}(f, \mathcal{C}_i) = k$，因此使得 $g_k(x(i)) = 0$）。

  因此，$f' = f \vee \overline{g_k}$ 的值在前 $p-1$ 个输入处变为 1。由于我们知道 $f(x^{(p)}) = 1$，且 $\mathsf{alt}(f, \mathcal{X}^*) = k$，可得 $\mathsf{alt}(f', \mathcal{X}^*) = k-1$。我们还可以证明，不存在交替次数超过 $k-1$ 且相对于 f' 的链。因此，$\mathsf{alt}(f') = k-1$。

  作为 $\mathsf{alt}(f') < k$，根据归纳假设，我们得到具有蕴含性质的函数 g'_1、g'_2、……、g'_{k-1} 以及 $f' = \neg(g'_1 \oplus g'_2 \oplus \cdots \oplus g'_{k-1})$。

- $\forall i \in [k-1]$，$g'_i \equiv g_i$: 如果某个输入 x 满足 $g_k(x) = 0$，这意味着 $a_f(x) = k$，从而对于 $i \in [k-1]$，$g_i(x) = 0$。在上面的部分中请注意，当 $\mathsf{alt}(f, \mathcal{X}) = k$ 时，我们证明了 $\mathsf{alt}(f', \mathcal{X}) = k-1$。这意味着对于所有 $i \in [k-1]$，$g'_i(x) = 0$ 也成立。

  另一方面，对于满足 $g_k(x) = 1$ 的输入 x，注意到 $f'(x) = f(x)$。由于 $a_f(x)$ 和 $a_{f'}(x)$ 仅取决于相应函数在 x “之上”的值，而函数 g_i 和 g'_i 正是基于这些值定义的，因此它们必然相等。

  因此，对于所有 x，我们都有 $g'_i(x) = g_i(x)$。根据 g_k 的定义，可以证明 $f \Rightarrow g_k$ 是一个重言式。结合这一事实与 $f' = f \vee \overline{g_k}$，可得 $f \equiv \overline{f'} \oplus g_k = g'_1 \oplus \ldots g'_{k-1} \oplus g_k = g_1 \oplus \cdots \oplus g_k$。

□

A.2 AC^0 有序输入的反相器构造——改编自 [18]

证明。我们详细说明定理 1.3 的证明中所用的构造。我们用更简单的记号和设定介绍 [18] 提出的构造。将 s 划分为 $t = n^\epsilon$ 个连续块 B_1、B_2、……、B_t，每块长度为 $p := m/n^\epsilon = \mathcal{O}(n^{k-\epsilon})$。注意，块 B_i 的否定具有 1^p 或 0^p 或 $1^j 0^{p-j}$ 的形式，其中某个 $0 \le j \le p$ 取决于 B_i 是否见证了 s 中从 0 到 1 的切换。我们的目标是构造一个 AC^0 电路，仅使用 $\mathcal{O}(n^\epsilon)$ 次取反。

令 B_r 表示发生开关切换的那个索引所在的块。对于给定的 s，称这样的块为特殊块。注意，这样的块至多有一个。对于每个块 B_i，定义 $b_i = B_i[1] \oplus B_i[p]$ ——即块 B_i 的首位和末位比特的奇偶性。注意，比特 b_i 表示 B_i 是否为特殊块。因此，我们可以按如下方式得到特殊块 $B_r = \vee_{i=1}^t ((b_i)^p \wedge B_i)$，其中 $\wedge$ 和 $\vee$ 按位进行，而 $(b_i)^p$ 是重复 p 次的比特 b_i。

得到特殊块 B_r的比特后，我们直接将上文生成的连线所承载的其中每个比特分别取反，得到 $B_{neg} := \overline{B_r}$。剩下要做的是识别每个块 B_i 的类型，并适当地连接连线，从而得到 1^p 或 0^p 或 B_{neg} 的取反结果。

具体做法如下： B_i 的取反为 $((b_i)^p \wedge B_{neg}) \vee ((\overline{b_i})^p \wedge (\overline{c_i})^p)$，其中 c_i 是 B_i的首位。为验证这一点，注意到上述表达式的输出在 $b_i = 1$时为 B_{neg}；否则，在 $c_i = 1$时为0^p，在 $c_i = 0$时为 1^p。所用取反的数量为 $\mathcal{O}(n^\epsilon) + \mathcal{O}(n^{k-\epsilon})$，第一部分用于生成$\overline{b_i}$和 $\overline{c_i}$，第二部分用于生成 B_{neg}；所有这些都可以在所有块之间共用连线。若 $\epsilon > \frac{k}{2}$，则问题解决。否则，我们已将取反数量降至 $\mathcal{O}(n^{k-\epsilon})$。为了进一步降低指数，我们递归地应用这一构造来对$B_r$取反，而根据定义，$B_r$仍然是有序的。这样，指数加法式地降低了 ϵ，而每一层只需增加常数深度。因此，经过 $\lceil k/\epsilon \rceil$ 层，我们将得到一个只需 $\mathcal{O}(n^\epsilon)$ 次取反的反相器。电路的规模关于 n 是多项式级的，因为每一层引入的门数都仅与有序比特的数量呈线性关系。

□