Condensing Hardness in Boolean Functions

Gabriel Hart
Advised by Kaave Hosseini and Alex Iosevich*

April 2025

1 Complexity Measures and Condensation

Let M be an $n \times n$ matrix over $\mathbb{R}$. If M is of rank r, then there exists an $r \times r$ submatrix of M with rank r. This is an example of a *condensation property* for the rank function. The rank function maps matrices over $\mathbb{R}$ to non-negative integers. For an input M of size n, there exists a restriction of the rank function to a subset of $M' \subseteq M$, whose size depends only on $\text{Rk}(M)$, and not on n, such that $\text{Rk}(M') = \text{Rk}(M)$. A *boolean function* is a function $f : \{0,1\}^n \to \{0,1\}$. A complexity measure C maps boolean functions to values in $\mathbb{N}$ (or sometimes in $\mathbb{R}^+$). A *subfunction* g of a boolean function f is a restriction of f to a subcube of the $\{0,1\}^n$ cube. That is, the values of some $(x_i)_{i \in S}$ are fixed, to obtain a function on the remaining $(x_j)_{j \in [n] \setminus S}$.

Every boolean function $f : \{0,1\}^n \to \{0,1\}$ can be represented by a unique multilinear polynomial P over $\mathbb{R}^n$. When evaluated at $x \in \{0,1\}^n \subset \mathbb{R}^n$, P takes values in $\{0,1\}$. It is important to note that if $x \in \{0,1\}^n$, any x_i in a monomial of P can be raised to any non-zero exponent without changing the value of $P(x)$. If a non-linear polynomial Q has the property that $Q(x) = f(x) \forall x \in \{0,1\}^n$, a multilinear polynomial P for f can be obtained from Q by replacing all exponents in Q with 1, and combining like terms. The number of non-constant monomials in P with non-zero coefficients gives a complexity measure called the *monomial sparsity* of f, denoted $\mathsf{Mon}(f)$. Each term in the multilinear polynomial for f is of the form

$$M(x) = \alpha_M \prod_{i \in S_M} x_i$$

where $\alpha_M \in \mathbb{R} \setminus \{0\}$, and $S_M \subseteq [n]$. Call the collection of sets $\{S_M\}$ for all monomials M of f the *monomial set system* for f. A *hitting set* for f is a set $H \subseteq [n]$ such that $H \cap S_M \neq \emptyset$ for every S_M in the monomial set system for f. The size of the smallest possible hitting set for the monomial set system of f gives a complexity measure called the hitting set size of f, denoted $\mathsf{HS}(f)$.

A boolean function f can also be represented by a *decision tree*. A decision tree T is a directed binary tree, with edges orient from root to leaf. Every internal vertex v of T is associated with a coordinate $i_v \in [n]$, and each leaf has a label in $\{0,1\}$. To evaluate f using T, start at the root of T. At vertex v, if $x_{v_i} = 1$, move to the right child of v, and if $x_{v_i} = 0$, move to the left child. Once a leaf is reached, f is given by the label of that leaf. The smallest depth of any decision tree that expresses f is a complexity measure called the *decision tree depth* of f, denoted $\mathsf{DT}(f)$. An AND decision tree functions the same as a decision tree, but each internal vertex is associated with a non-empty test set $M_v \subseteq [n]$. At vertex v in an AND decision tree, the monomial $\prod_{i \in M_v} x_i$ is evaluated, or equivalently, the boolean AND $\bigwedge_{i \in S_v} x_i$ is evaluated, to determine which branch is taken. The minimum depth of AND decision tree required to represent f is denoted $\mathsf{AND}(f)$.

The complexity measures discussed so far have the following relationships:

$$\mathsf{HS}(f) \leq \mathsf{AND}(f) \leq \mathsf{Mon}(f)$$

*Thanks also go to Zahra Hadizadeh for her thoughts and contributions in the early stages of this project

The bound $\mathsf{HS}(f) \leq \mathsf{Mon}(f)$ is immediate from the fact that choosing one x_i from each monomial of f gives a hitting set for f. To see that $\mathsf{AND}(f) \leq \mathsf{Mon}(f)$, first, observe that once the value of every monomial of f at x is known, the value of $f(x)$ is uniquely determined. Let T be an AND decision tree of depth $\mathsf{Mon}(f) = k$, with every layer full. Let $M_1 ... M_k \subseteq [n]$ be the monomials of f. Equip every node in layer j of T with the test monomial M_j. Any path from the root of T to a leaf of T evaluates every monomial of T. If the evaluation paths for x, y terminate in the same leaf of T, then for each monomial M_j, $M_j(x) = M_j(y)$, so $f(x) = f(y)$. Labeling each leaf of T by $f(x)$ for an arbitrary x in that leaf gives an AND decision tree for f of depth k.

Showing $\mathsf{HS}(f) \leq \mathsf{AND}(f)$ requires slightly more work. A monomial M_i of f is called *minimal* if there is no monomial M_j of f such that $M_i \subset M_j$. If M_j is not a minimal monomial, then f necessarily has another monomial $M_i \subset M_j$. Therefore, $H \subseteq [n]$ is a hitting set of f iff H has non-empty intersection with every minimal monomial of f. Let M_i be a minimal monomial for f, and $x \in \{0,1\}^n$ be defined by $x_j = 1$ iff $j \in M_i$. Then $M_i(x) = 1$, and $M_j(x) = 0 \forall j \neq i$. Since f is a boolean function and M_i has a non-zero coefficient, the coefficient on M_i in the polynomial for f must be 1 or -1. Importantly, all monomials $M_j, j \neq i$ have the same value at x as at 0. Only the value of M_i differs. We can thus conclude $f(x) \neq f(0)$. Let $\{M_i\}_{i \in J}$ be the minimal monomials of f. Define $y \in \{0,1\}^n$ by $y_i = 1$ iff $i \in \bigcup_{i \in J} M_i \setminus H$. Then $f(y) = f(0)$, since every monomial of f takes on the same value at y as at 0. However, if $k \in H \cap M_i$, where M_i is a minimal monomial, then $M_i(y + e_k) \neq M_i(y)$, but for all $j \neq i$, $M_j(y + e_k) = M_j(y) = M_j(0)$. Thus, at y, f is *sensitive* to all coordinates in H, in that flipping any coordinate in H flips the value of f. Let $M_{y_1} ... M_{y_r}$ be the sequence of monomials corresponding to the vertices in an AND decision tree for f visited when evaluating $f(y)$. If there exists a coordinate $k \in H$ such that $x_k \notin M_{y_i}$ for any i, then we would have $f(y + e_k) = f(y)$, so $\bigcup_{i=1}^{r} M_{y_i}$ must cover H. Suppose $k \in H$ is such that if $k \in M_{y_i}$ for some i, then there exists $k' \neq k \in H \cap M_{y_i}$. Then again we have $f(y + e_k) = f(y)$, since no monomials M_{y_i} differ between $y + e_k$ and y. Therefore, the evaluation path for f at y contains at least one monomial for every $k \in H$, and we can therefore conclude

$$\mathsf{AND}(f) \geq r \geq |H| = \mathsf{HS}(f)$$

Additionally, the following bounds hold:

$$\mathsf{AND}(f) \leq \mathsf{DT}(f) \leq n$$

This is because every decision tree is also an AND decision tree. In particular, an AND decision tree is a decision tree if the test set at each vertex contains only a single coordinate. $\mathsf{DT}(f) \leq n$ follows by exactly the same argument as $\mathsf{AND}(f) \leq \mathsf{Mon}(f)$. If all n coordinates of x are known, then the value of f can be determined.

2 Condensing Monomial Sparsity

This section will prove the following condensation theorem for monomial sparsity:

Theorem 1 *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with $\mathsf{Mon}(f) = k$. Then there exists a subfunction g of f on $O(k^2)$ coordinates with $\mathsf{Mon}(g) = k$.*

The theorem is trivially true when $k \geq \sqrt{n}$, since in that case, taking $g = f$ is sufficient. Since $\mathsf{Mon}(f)$ can be as high as 2^n, Theorem 1 is only relevant in the particular case of low monomial sparsity. The proof of Theorem 1 follows from a more general combinatorial lemma about set systems.

2.1 Isomorphic Set Systems

Let X be any set. A *set system* S over X is a collection of non-empty elements of the power set 2^X. If S_1 is a set system over X_1 and S_2 is a set system over X_2, define a *set system isomorphism* to be a bijection

$h : S_1 \to S_2$ that preserves inclusions and pair-wise disjointness. That is, $A \subseteq B$ iff $h(A) \subseteq h(B)$, and $A \cap B = \emptyset$ iff $h(A) \cap h(B) = \emptyset$.

Lemma 1 *Let S be a set system over a set X, with $|S| = k$. Then there exists $A \subset X$ with $|A| \leq \frac{3k(k-1)}{2}$ such that the set system $\{E \cap A : E \in S\}$ is isomorphic to S. An isomorphism $h : S \to \{E \cap A : E \in S\}$ is given by $h(E) = E \cap A$.*

To prove this lemma, we will construct such a set A. For each pair of sets $E_i, E_j \in S$, include in A one element from each from $E_i \cap E_j$, $E_i \setminus E_j$, and $E_j \setminus E_i$, provided the set in question is non-empty.

For a pair a given pair of sets $E_i, E_j \in S$, at least one, and at most all three of these will be non-empty. There are $\frac{k(k-1)}{2}$ pairs of sets E_i, E_j, and each contributes at most 3 distinct elements to A, so $|A| \leq \frac{3k(k-1)}{2}$. What remains is to show that $h : S \to \{S \cap A : E \in S\}$ given by $h(E) = E \cap A$ is a set system isomorphism.

First, suppose E_i, E_j are disjoint sets in S. Since $A \cap E_i \subseteq E_i$ and $A \cap E_j \subseteq E_j$, $h(E_i) \cap h(E_j) = \emptyset$. Conversely, if $h(E_i) \cap h(E_j) \neq \emptyset$, then there exists $a \in A$ such that $a \in E_i \cap E_j$. Then $a \in A \cap E_i$ and $a \in A \cap E_j$, so $h(E_i) \cap h(E_j) \neq \emptyset$.

Next, suppose $E_i \subseteq E_j$. Then $E_i \cap A \subseteq E_j \cap A$, so $h(E_i) \subseteq h(E_j)$. If $h(E_j) \not\subseteq h(E_i)$, then there exists a such that $a \in A \cap E_j$ but $a \notin A \cap E_i$. Therefore, $(A \cap E_i) \setminus (A \cap E_j) \neq \emptyset$. But $(A \cap E_i) \setminus (A \cap E_j) \subseteq E_i \setminus E_j$, so $E_i \setminus E_j \neq \emptyset$, and therefore $E_i \not\subseteq E_j$. $\square$

2.2 Proof of Theorem 1

Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function given by a polynomial P, and let $M_1 ... M_k$ be the monomials of P. The monomials of P form a set system $M = \{M_1 ... M_k\}$ over $[n]$. By Lemma 1, there exists $A \subseteq [n]$ with $|A| \leq \frac{3k(k-1)}{2}$ such that the set system $M' = \{M_j \cap A : M_j \in M\}$ is isomorphic to M. Let g be the subfunction of f obtained by fixing x_i to 1 for all $i \notin A$. In general, fixing x_i to 0 for some i causes every monomial of P containing x_i to become identically 0. Fixing x_i to 1 has the effect of removing x_i from each monomial in which it occurs. Since g is defined by fixing coordinates only to 1, and not to 0, a polynomial Q for g can be obtained by removing x_i from each monomial M_j of P, and then combining like terms.

The monomials obtained by removing all $x_i \notin A$ from the monomials of P are precisely $M' = \{M_j \cap A : M_j \in M\}$. P is already assumed to be fully simplified, so if M_i, M_j are distinct monomials of P, then at least one of $M_i \setminus M_j$ and $M_j \setminus M_i$ is non-empty. Because M' is isomorphic to M under the set system isomorphism mapping $M_i \mapsto M'_i$, at least one of $M'_i \setminus M'_j$ and $M'_j \setminus M'_i$ is non-empty. Therefore, no terms combine in Q, and the sets in M' are precisely the monomials of Q. $\square$

Remark: Q is obtained merely by removing some x_i from the monomials of P. No terms combine, and no modification is made to the coefficients on the monomials. Therefore, each monomial M'_i has exactly the same coefficient in Q as M_i does in P. Put another way, the set system isomorphism $M_i \mapsto M'_i$ preserves the coefficients on monomials when considered as a map from the terms of P to the terms of Q.

2.3 Sharpness

At the time of writing, the question remains open as to whether the $O(k^2)$ condensation given in Theorem 1 is the best possible condensation for monomial sparsity, up to constants. The following example, however, provides a lower bound of $\Omega(k)$.

Define $f : \{0,1\}^n \to \{0,1\}$ by $f(x) = 1$ iff $x = 1 - e_i$ for some i. That is, $f(x) = 1$ iff there is some i such that $x_i = 0$, and $x_j = 1 \forall j \neq i$. The multilinear polynomial for f is given by

$$P(x) = \left(\sum_{i=1}^{n} \prod_{j \neq i} x_j \right) - n \cdot \prod_{j=1}^{n} x_j$$

This expression for P is fully simplified, and we can observe that $\mathsf{Mon}(f) = n+1$. Suppose we wished to build a subfunction g of f with $\mathsf{Mon}(g) = \mathsf{Mon}(f)$. Fixing any x_i to 0 is out of the question. For any x_i, the only monomial not containing x_i is $\prod_{j \neq i} x_j$. Fixing x_i to 0 causes all other monomials to become identically 0, and the resulting function will have a monomial sparsity of 1. On the other hand, fixing x_i to 1 results in x_i being removed from each monomial of P. In this case, the monomial $M_1 = \prod_{j \neq i} x_i$ remains unchanged, while the monomial $M_2 = \prod_{j=1}^{n} x_j$ becomes equal to M_1, and M_1 and M_2 combine into a single monomial. In particular, the subfunction g obtained by fixing $x_i = 1$ in f has $\mathsf{Mon}(g) = \mathsf{Mon}(f) - 1$ and is given by the polynomial

$$Q((x_j)_{j \neq i}) = \left(\sum_{i \neq j} \prod_{k \neq i,j} x_k \right) - (n-1) \prod_{j \neq i} x_j$$

3 Condensing Hitting Set Size

The notions of minimal monomials and hitting sets as defined previously for the monomial set system of a boolean function's polynomial can be extended to arbitrary set systems. One property worth noting is that a set system isomorphism $h : S_1 \to S_2$ preserves minimal sets. If $E \in S_1$ is such that there does not exist $F \in S_1 \setminus \{E\}$ with $F \subseteq E$, then the same will be true of $h(E)$ in S_2 due to the inclusion preserving property of set system isomorphisms. Conversely, if $F \subseteq E$, then $h(F) \subseteq h(E)$. As noted previously, a necessary and sufficient condition for $H \subseteq [n]$ to be a hitting set for the monomial set system M of a boolean function f is for H to have non-empty intersection with every minimal monomial in M. Given the positive result for condensation of monomial sparsity, it is reasonable to ask whether hitting set size would condense by a similar construction.

3.1 Condensing With Respect to Monomial Sparsity

One result for condensation of hitting set size comes as a corollary of Theorem 1:

Corollary 1 *Let $f : \{0,1\}^n$ be a boolean function with $\mathsf{Mon}(f) = k$ and $\mathsf{HS}(f) = r$. Then there exists a subfunction g of f on $O(k^2)$ variables with $\mathsf{HS}(g) \geq r$.*

Let g be the same subfunction constructed in the proof of Theorem 1, and M and M' be the monomial set systems for f and g respectively. Let H be a hitting set of minimum possible size for M'. By construction, $M'_i \subseteq M_i$. Therefore, if $H \cap M'_i \neq \emptyset$, then $H \cap M_i \neq \emptyset$. Thus, H is also a hitting set for M, though not necessarily an optimal one.

Remark: It should be noted that in general, set system isomorphisms do not preserve hitting set size. As defined, our notion of set system isomorphism captures only the pair-wise incidence relations between elements of the set system, and does distinguish structural differences that require more than 2 elements to observe. For example, the following set systems are isomorphic:

$$S_1 = \{\{1,2\},\{2,3\},\{3,1\}\}$$
$$S_2 = \{\{1,2\},\{1,3\},\{1,4\}\}$$

However, $\{1,2\}$ is an optimal hitting set for S_1, whereas $\{1\}$ is an optimal hitting set for S_2.

3.2 Restricting to a Hitting Set Fails

Corollary 1 provides a condensation of hitting set size to a subfunction of size dependent on monomial sparsity. The question remains, though, as to whether hitting set size can be condensed in a subfunction of size depending on $\mathsf{HS}(f)$. One possible method could be to take a minimum hitting set H of f, and fix x_i to 1 for all $i \notin H$. This construction would seem to preserve every minimal monomial, but for the fact that it may produce like terms, which might then combine to 0.

The following examples illustrate the failure of this approach. Consider the boolean function

$$f(x, y, z) = xy + xz - 2xyz$$

An optimal hitting set for f is $H = \{x\}$. Fixing y, z to 1 results in the subfunction

$$f(x, 1, 1) = x + x - 2x \equiv 0$$

More significantly, this function f can be used as a building block to construct larger examples. Let $g : \{0,1\}^n \to \{0,1\}$ be the indicator function of the 0 vector. A polynomial expressing g is given by

$$g(x) = \prod_{i=1}^{n}(1 - x_i)$$

Note that every possible monomial occurs with non-zero coefficient in g. Define $h : \{0,1\}^{3n} \to \{0,1\}$ by

$$h(x_1, y_1, z_1 ... x_n, y_n, z_n) = g(f(x_1, y_1, z_1), ... f(x_n, y_n, z_n))$$

From the polynomial representations of f and g, we know that $\forall i$, the monomials $x_i y_i$ and $x_i z_i$ occur in h with coefficient -1. Therefore, a hitting set for h must hit all such monomials. Specifically, $H = \{x_1 ... x_n\}$ is a minimum hitting set for h. Fixing all variables not in H to 1 gives the subfunction

$$g(f(x_1, 1, 1) ... f(x_n, 1, 1)) = g(0, ... 0) \equiv 1$$

To summarize, we have constructed a boolean function h on $3n$ variables such that when all coordinates outside of a minimum hitting set are fixed to 1, the resulting subfunction is constant.

4 Condensing AND Decision Tree Depth

In this section, I will prove the following condensation theorem[1] for AND decision tree depth:

Theorem 2 *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with $\mathsf{Mon}(f) = k$. Then there exists a subfunction g of f on $O(k^2)$ variables with $\mathsf{AND}(g) \geq \mathsf{AND}(f)$. Moreover, g is the same subfunction as the condensation of monomial sparsity constructed in Theorem 1.*

Note that the size of the condensation is still a function of $\mathsf{Mon}(f)$, rather than $\mathsf{AND}(f)$. [**Note: this should not yet be called a theorem, as it turns out the proof is incorrect. I have left the incorrect proof in this draft so you can see what I'm trying to do and if there's any way to salvage it.**]

4.1 Maximal Monomials

Recall that a monomial M_i of a boolean function f is considered maximal iff there do not exist any monomials M_j of f with $M_i \subset M_j$. The proof of theorem 2 will require the following lemma about maximal monomials: [**note: this lemma is no longer used in the proof of theorem 2, but I do plan to still include it**]

Lemma 2 *Every boolean function has a unique maximal monomial.*

[1] Not actually proved yet, should say conjecture.

Let $f : \{0,1\}^n \to \{0,1\}$. The lemma holds trivially in the case that f is constant or $\mathsf{Mon}(f) = 1$, so we can assume f has at least 2 distinct monomials. By way of contradiction, suppose f has exactly 2 maximal monomials, M_1 and M_2. Define $g_1, g_2, g_{1,2}$ to be the subfunctions obtained by fixing the coordinates in $M_2 \setminus M_1$, $M_1 \setminus M_2$, and $M_1 \Delta M_2$ to 0 respectively. The monomials for g_1 are precisely those monomials M of f with $M \subseteq M_1$. Similarly, the monomials of g_2 are those with $M \subseteq M_2$, and the monomials of $g_{1,2}$ have $M \subseteq M_1 \cap M_2$. In particular, extending $g_1, g_2, g_{1,2}$ to the whole space $\{0,1\}^n$, we can write

$$f(x) = g_1(x) + g_2(x) - g_{1,2}(x)$$

Define $X_1, X_2 \subseteq \{0,1\}^n$ by

$$X_1 = \{x \in \{0,1\}^n : g_1(x) = 1 - g_{1,2}(x)\}$$
$$X_2 = \{x \in \{0,1\}^n : g_2(x) = 1 - g_{1,2}(x)\}$$

If $X_1 \cap X_2 = \emptyset$, then $g_1 \equiv 1 - g_2$, and

$$f(x) = (1 - g_2(x)) + g_2(x) - g_{1,2}(x) = 1 - g_{1,2}(x)$$

In particular, this implies that the maximal monomials of f are precisely the maximal monomials of $g_{1,2}$. But all monomials M of $g_{1,2}$ have $M \subseteq M_1 \cap M_2$, contradicting the assumption that M_1 and M_2 are, themselves, distinct maximal monomials of f.

On the other hand, suppose $X_1 \cap X_2 \neq \emptyset$. Then there exists $x \in \{0,1\}^n$ such that $g_1(x) = g_2(x) = 1 - g_{1,2}(x)$. Then $f(x) = 2 - 3g_{1,2}(x)$. If $g_{1,2}(x) = 0$, then $f(x) = 2$. If $g_{1,2} = 1$, then $f(x) = -1$. In either case, the premise that f is a boolean function is contradicted.

What remains is the case where f might have more than 2 maximal monomials. Suppose f has maximal monomials $M_1 ... M_r$. Fixing all x_i to 1 for $i \in \bigcup_{j=1}^{r} M_j \setminus (M_1 \cup M_2)$ gives a subfunction g which, if not constant, has the maximal monomials M_1 and M_2. Since M_1 and M_2 are each maximal, $M_1 \setminus M_2$ and $M_2 \setminus M_1$ are both non-empty, so g is not constant. However, the existence of such a g is disallowed by the above result. □

Corollary 2 *Let $f : \{0,1\}^n$ be a boolean function with monomials $\{M_i\}_{i \in [k]}$, given by an optimal AND decision tree T. If N is the test monomial for a vertex of T, then $N \subseteq M_i$ for some monomial M_i of f.*

The corollary is almost immediate from the observation that a coordinate x_i appears in some node of T iff x_i appears in some monomial M_i for f. If f is constant with respect to x_i, then fixing x_i to 0 does not change the value of f. This does, however, kill any monomials and AND decision tree nodes that contain x_i, creating a subfunction identical to f when extended to the whole $\{0,1\}^n$ cube, but with better monomial sparsity and/or AND decision tree depth. However, T is assumed to be optimal, and the polynomial representation of f is unique, so this is not possible. If f is not constant on x_i, then there exists $x \in \{0,1\}^n$ such that $f(x) \neq f(x + e_i)$. If x_i does not occur in any monomial of f, then we would have $f(x) = f(x + e_i)$. If x_i is not tested in any node N of T, then T also cannot distinguish between x and $x + e_i$. By the previous lemma, there exists a maximal monomial M_0 of f such that $M_i \subseteq M_0$ for every monomial M_i of f. That is, if x_j is present in any monomial of f, then M_0 contains x_j. Thus, for every test monomial N in T, we have $N \subseteq M_0$.

4.2 The Topology of AND Decision Trees

Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with monomials $\{M_1 ... M_k\}$, represented by an AND decision tree T. Define an equivalence relation $\sim$: $[n] \times [n] \to \{0,1\}$ by $x \sim y$ iff for all M_i, $x \in M_i \Leftrightarrow y \in M_i$. The equivalence classes of $\sim$ form a basis $\mathcal{B}$ for a topology $\mathcal{T}$ on $[n]$, which is the same as the topology induced by the sub-basis of $\{M_1 ... M_k\}$.

Lemma 3 *Define an AND decision tree T' by, for each node v in T, replace the test monomial N at v with a minimal open set containing N in the topology $\mathcal{T}$.*[2] *T' is a minimal AND decision tree that decides f.*

A minimal open cover relative to $\mathcal{T}$ for a set $N \subseteq [n]$ can be obtained by taking $\bigcup_{j \in N} \beta_j$, where $\beta_j \in \mathcal{B}$ is the basis element containing j. Each j is covered by a unique basis element β_j, since the basis $\mathcal{B}$ consists of the equivalence classes of an equivalence relation on $[n]$. To prove the lemma, it will suffice to show that replacing a single x_j by $(x_i)_{i \in \beta_j}$ results in a tree that still decides f. Replacing each x_j one at a time in this manner inductively gives the desired result. Moreover, we only need to show the replacement of x_i and x_j $x_i x_j$ where $i \sim j$, since the result follows inductively for larger equivalence classes.

Choose some $i \sim j \in [n]$, and let T_1 be the AND decision tree that is identical to T except that every test monomial N in T with $x_i \in N$ or $x_j \in N$ has been replaced by the linear monomial N' equal to $N x_i x_j$ on the $\{0,1\}^n$ cube. Suppose x has $x_i = x_j = 1$. Then $x_i x_j = 1$, and for every test monomial, we have $N(x) = N'(x)$. Therefore, x has the same evaluation path in T_1 as in T, and T_1 correctly classifies x. Similarly, if $x_i = x_j = x_i x_j = 0$, x has the same evaluation path in T and T_1. If one of $x_i = 0$ and $x_j = 0$ holds, then $x_i x_j = 0$. Let y be identical to x, except $y_i = y_j = 0$. The evaluation path of x in T_1 is the same as that of y in T_1, which in turn is identical to the evaluation path for y in T. However, this is fine, because if $i \in M$ for some monomial M, then $j \in M$ as well, so $M(x) = M(y)$. If a monomial M is such that $i, j \notin M$, then $M(x) = M(y)$ since x and y are identical at coordinates other than i, j. Since all monomials of f take the same value on x and y, $f(x) = f(y)$. Therefore, x is still classified correctly by T_1. $\square$

To summarize, every boolean function f has an optimal AND decision tree where every node examines a set of coordinates that is open in the topology on $[n]$ generated by the sub-basis of the monomials of f.

4.3 Proof of Theorem 2

Recall the construction used in Theorem 1 to condense monomial sparsity. Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with $\mathsf{Mon}(f) = k$, and let $g : A \to \{0,1\}$ be the condensation found in Theorem 1, where $A \subseteq [n]$ is the set of coordinates not fixed by g, and $|A| = O(k^2)$. Let $s : \{M_1 ... M_k\} \to \{M'_1 ... M'_k\}$ be the set system isomorphism from the monomials of f to the monomials of g. Lemma 3 guarantees the existence of an optimal AND decision tree T' for g where the test sets at the vertices of T' are open in the topology on A generated by the sub-basis $\{M'_1 ... M'_k\}$. To prove Theorem 2, we will build an AND decision tree T for f that is isomorphic as a graph to T'.

Let $\mathcal{T}'$ be the quotient topology on A induced by $\sim$, which, as established previously, is the same as the topology generated by the sub-basis $\{M'_1 ... M'_k\}$. Every test set N' of a vertex v' in T' is open in $\mathcal{T}'$. By definition of the topology generated by a sub-basis, we can write $N' = \bigcup_{i \in I} \bigcap_{j \in J_i} M'_j$. Let the corresponding vertex in T have the test monomial $N = \bigcup_{i \in I} \bigcap_{j \in J_i} M_i$ where $s(M_i) = M'_i$. Note that intersecting the test monomials of T with A gives T' exactly.

Since determining the value of all monomials M_i at a point x is sufficient to determine $f(x)$, if $y = \bigcup_{M_i(x)=1} M_i$, then $f(x) = f(y)$. It is therefore sufficient to prove that T decides f at just those x for which $\{i : x_i = 1\}$ can be given as a union of monomials M_i. Suppose x is such that $\{i : x_i = 1\} = \bigcup_{i \in I} M_i$, and let z be the corresponding restriction of x to the coordinates in A. Then

$$z = \bigcup_{i \in I} M_i \cap A = \bigcup_{i \in I} M'_i$$

[2] The notion of a minimal open set that contains an arbitrary set is not well-defined in general topological spaces, but our topology $\mathcal{T}$ has finitely many open sets.

Thus, for all i, we have $M_i(x) = M'_i(z)$. Recall that the polynomial representations for f and g are given by

$$f(x) = \sum_{i=1}^{k} \alpha_i M_i(x)$$
$$g(z) = \sum_{i=1}^{k} \alpha_i s(M_i)(z) = \sum_{i=1}^{k} \alpha_i M'_i(z)$$

Since for all i, $M_i(x) = M'_i(z)$, we have $f(x) = g(z)$. Also,

$$A \cap \bigcup_{i \in I} \bigcap_{j \in J_i} M_j = \bigcup_{i \in I} \bigcap_{j \in J_i} A \cap M_j = \bigcup_{i \in I} \bigcap_{j \in J_i} M'_j$$

Therefore, for corresponding decision nodes N and N' in T and T', we have $N(x) = N'(z)$[3], so the evaluation path of x in T is the same as the evaluation path of z in T', meaning T decides f. T' is an optimal AND decision tree for g, and T and T' have the same depth, so $\mathsf{AND}(f) \leq \mathsf{AND}(g)$. $\square$

References

[Hru24] Pavel Hrubeš. "Hard Submatrices for Non-Negative Rank and Communication Complexity". In: *39th Computational Complexity Conference (CCC 2024)*. Ed. by Rahul Santhanam. Vol. 300. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 13:1–13:12. ISBN: 978-3-95977-331-7. DOI: 10.4230/LIPIcs.CCC.2024.13. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CCC.2024.13.

[Kno+21] Alexander Knop, Shachar Lovett, Sam McGuire, et al. "Log-rank and lifting for AND-functions". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2021. Virtual, Italy: Association for Computing Machinery, 2021, pp. 197–208. ISBN: 9781450380539. DOI: 10.1145/3406325.3450999. URL: https://doi.org/10.1145/3406325.3450999.

[Yan91] Mihalis Yannakakis. "Expressing combinatorial optimization problems by Linear Programs". In: *Journal of Computer and System Sciences* 43.3 (1991), pp. 441–466. ISSN: 0022-0000. DOI: https://doi.org/10.1016/0022-0000(91)90024-Y. URL: https://www.sciencedirect.com/science/article/pii/002200009190024Y.

[LS88] L. Lovasz and M. Saks. "Lattices, mobius functions and communications complexity". In: *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*. 1988, pp. 81–90. DOI: 10.1109/SFCS.1988.21924.

[Göö+24] Mika Göös, Ilan Newman, Artur Riazanov, et al. "Hardness Condensation by Restriction". In: *Electronic Colloquium on Computational Complexity*. Weizmann Institute of Science, 2024. URL: https://eccc.weizmann.ac.il/report/2023/181/.

[KSP20] Rohan Karthikeyan, Siddharth Sinha, and Vallabh Patil. "On the resolution of the sensitivity conjecture". In: *Bulletin of the American Mathematical Society* 57.4 (Oct. 2020), pp. 615–638. ISSN: 0273-0979. DOI: 10.1090/bull/1697.

[3] This is the incorrect step in the proof.