# Condensing Hardness in Boolean Functions

Gabriel Hart
Advised by Kaave Hosseini and Alex Iosevich[*]

April 2025

**Abstract**

A complexity measure on boolean functions $f : \{0,1\}^n \to \{0,1\}$ condenses if the complexity of $f$ can be witnessed by some restriction of $f$ to a number of variables depending only on the complexity of $f$. In this paper, I explore the condensation properties of AND decision tree depth and monomial sparsity, and connections to communication complexity and the log-rank conjecture of Lovasz and Saks. The main result introduced is a condensation theorem for monomial sparsity, along with an approximate sharpness example for this result. Additionally, I investigate the topological properties of the monomials of the linear polynomial representing a boolean function and the coordinate sets tested in that function's AND decision tree.

## 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 $\mathrm{Rk}(M)$, and not on $n$, such that $\mathrm{Rk}(M') = \mathrm{Rk}(M)$. 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

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

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 $\prod\limits_{i \in M_v} x_i$ is evaluated, or equivalently, the boolean AND $\bigwedge\limits_{i \in S_v} x_i$ 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:

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

The bound $\mathsf{HS}(f) \le \mathsf{Mon}(f)$ is immediate from the fact that choosing one $x_i$ from each monomial of $f$ gives a hitting set for $f$. To see that $\mathsf{AND}(f) \le \mathsf{Mon}(f)$, 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 $\mathsf{HS}(f) \le \mathsf{AND}(f)$ 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 $M_j(x) = 0 \forall j \ne i$. 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 $M_j, j \ne i$ have the same value at $x$ as at 0. Only the value of $M_i$ differs. We can thus conclude $f(x) \ne f(0)$. Let $\{M_i\}_{i \in J}$ be the minimal monomials of $f$. Define $y \in \{0,1\}^n$ by $y_i = 1$ iff $i \in \bigcup\limits_{i \in J} M_i \setminus H$. 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 $M_i(y + e_k) \ne M_i(y)$, but for all $j \ne i$, $M_j(y + e_k) = M_j(y) = M_j(0)$. 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 $\bigcup\limits_{i=1}^{r} M_{y_i}$ must cover $H$. Suppose $k \in H$ is such that if $k \in M_{y_i}$ for some $i$, then there exists $k' \ne k \in H \cap M_{y_i}$. 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

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

Additionally, the following bounds hold:

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

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. $\mathsf{DT}(f) \le n$ follows by exactly the same argument as $\mathsf{AND}(f) \le \mathsf{Mon}(f)$. If all $n$ coordinates of $x$ are known, then the value of $f$ can be determined.

Finally, the upper bound $\mathsf{Mon}(f) \le 3^{\mathsf{AND}(f)}$ is known. For proof, the reader is referred to [Kno+21]. Here, I will prove the slightly weaker bound $\mathsf{Mon}(f) \le 4^{\mathsf{AND}(f)}$. Let $T$ be an AND decision tree of depth $m$ for the boolean function $f$, and let $g_1 ... g_k$ be the indicator functions for those leaves of $T$ labeled 1. That is, $g_j(x) = 1$ iff $f(x) = 1$ and the evaluation path of $x$ in $T$ terminates in the leaf corresponding to $g_j$. We can write $f(x) = \sum\limits_{j=1}^{k} g_j(x)$, since for any $x$, $g_j(x) = 1$ for at most one $j$. For each $g_j$, let $L_j$ be the set of all nodes for which the 0 branch is taken in the path to the leaf $g_j$, and $R_j$ be the set of all nodes where the right branch is taken. Then for each $j$,

$$g_j(x) = \left(\prod_{N \in L_j} (1 - N(x))\right)\left(\prod_{N \in R_j} N(x)\right)$$

If the path ending at $g_j$ is of length $m$, the above expression for $g_j$ will produce at most $2^m$ distinct monomials. An AND decision tree of depth $m$ has at most $2^m$ leaves, and the path to any leaf is of length at most $m$. Therefore, $f(x) = \sum_{j=1}^{k} \left(\prod_{N \in L_j} (1 - N(x))\right)\left(\prod_{N \in R_j} N(x)\right)$ has at most $2^m \cdot 2^m = 2^{2m} = 4^m$ monomials.

## 2 Communication Complexity

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Suppose Alice has $x \in \{0,1\}^n$, Bob has $y \in \{0,1\}^n$, and Alice wishes to compute $f(x,y)$. Alice and Bob are allowed to agree beforehand on a deterministic communication protocol, where one bit of information can be sent from one party to the other (in either direction), until Alice has computed $f$. Let $B(p,x,y)$ denote the number of bits of information transmitted when protocol $p$ is used to compute $f(x,y)$. Let $P$ denote the set of all possible deterministic communication protocols for $f$. The *deterministic communication complexity* of $f$ is defined as

$$\mathsf{CC}(f) = \min_{p \in P} \max_{x,y \in \{0,1\}^n} B(p,x,y)$$

For any function $f$, it is always possible for Bob to send all of $y$ to Alice, so for any choice of function $f$, $\mathsf{CC}(f) \le n$. An example of a function with low communication complexity is the parity function, defined by $\mathsf{Par}(x,y) = 1$ iff the total number of 1 coordinates in $x$ and $y$ is odd. For Alice to compute $\mathsf{Par}(x,y)$, she needs to know only whether Bob has an odd number of 1 coordinates in $y$, which Bob can communicate with a single bit of information. However, $\mathsf{Par}(x,y)$ still depends on both $x$ and $y$, so Alice cannot decide $\mathsf{Par}(x,y)$ based on $x$ alone, necessitating a non-zero amount of communication. In particular, the protocol described previously is best possible, and we have $\mathsf{CC}(\mathsf{Par}) = 1$.

A function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can also be described by a $2^n \times 2^n$ communication matrix $F$. The rows and columns of $F$ are indexed by $\{0,1\}^n$, and $F_{xy}$ is defined to be $f(x,y)$. Lovasz and Saks [LS88] proposed the following *log-rank conjecture*:

**Conjecture 1** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be represented by communication matrix $F$. Then for some positive universal constant $c$,*

$$\mathsf{CC}(f) = O(\log^c \mathrm{Rk}(F))$$

Part of the challenge in resolving this conjecture has come from its generality; the bound is conjectured to hold for any possible choice of $f$. As such, many efforts have focused on resolving the conjecture for smaller spaces of boolean functions.

Given a function $f : \{0,1\}^n \to \{0,1\}$, and a *gadget* function $g : \{0,1\}^2 \to \{0,1\}$, we can define a *lifted function* $f_g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ by

$$f_g(x,y) = f(g(x_1,y_1)...g(x_n,y_n))$$

One approach is to use lifting theorems, which, for a specific choice of gadget $g$, relate $\mathsf{CC}(f_g)$ to various complexity measures on $f$. Here, we will consider AND functions, which are lifted functions using the AND gadget, $x, y \mapsto x \wedge y$. The following lemma and proof from [Kno+21] relates the communication complexity of an AND function to that function's AND decision tree depth.

**Lemma 1** *Let $f : \{0,1\}^n \to \{0,1\}$ be any boolean function, and $f_\wedge : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ denote the lifted AND function. Then $\mathsf{CC}(f_\wedge) \le 2 \cdot \mathsf{AND}(f)$.*

The proof is to use the AND decision tree for $f$ to construct a communication protocol for $f_\wedge$. Consider a node $N$ in an AND decision tree $T$ for $f$, as used to compute $f_\wedge(x,y) = f(x \wedge y)$.

$$N(x \wedge y) = \prod_{i \in N} (x \wedge y)_i = \prod_{i \in N} x_i y_i = \left(\prod_{i \in N} x_i\right)\left(\prod_{i \in N} y_i\right) = N(x) \cdot N(y)$$

Therefore if Alice and Bob each privately compute $N(x)$ and $N(y)$, then exchange the results, they both gain access to $N(x \wedge y)$. Starting at the root of $T$, Alice and Bob can each transmit 1 bit, then take the same branch, repeating until a leaf is reached. At every node in the path followed, 2 bits of information are exchanged. Thus, we have a communication protocol for $f_\wedge$ that transmits $2 \cdot \mathsf{AND}(f)$ bits in the worst case choice of $x, y$. $\square$

The following result, also due to Knop et. al, relates AND decision tree depth to monomial sparsity:

**Theorem 1** *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with $\mathsf{Mon}(f) = k$. Then*

$$\mathsf{AND}(f) = O(\log^5(k) \cdot \log(n))$$

The reader is referred to [Kno+21] for a full proof. What I will draw our attention to are the consequences of this result for the log-rank conjecture. In conjunction with Lemma 1, it produces the bound

$$\mathsf{CC}(f_\wedge) \leq 2 \cdot \mathsf{AND}(f) = O(\log^5(\mathsf{Mon}(f)) \cdot \log(n))$$

This is very closed to the bound proposed by the log-rank conjecture, except for the $\log(n)$ factor. In fact, when $\mathsf{Mon}(f) = \Omega(\log^r(n))$ for some fixed $r$, the conditions of the log-rank conjecture are satisfied. It is only the case of very sparse functions that remains unresolved. This is where condensation enters the picture. Suppose we were to prove a condensation result of the following form:

**Conjecture 2** *Let $f : \{0,1\}^n \to \{0,1\}$ has $\mathsf{Mon}(f) = k$. Then there exists a sub-function $g$ of $f$ on $P(k)$ variables such that $\mathsf{AND}(g) \geq \mathsf{AND}(f)^c$ for some universal constant $c$, $0 < c < 1$.*

Assuming this conjecture, Theorem 1 gives

$$\mathsf{AND}(f)^c = O(\log^5 \mathsf{Mon}(f) \cdot \log P(\mathsf{Mon}(f)))$$

In particular, suppose $P(k) = 2^{\log^\alpha(k)}$. Then

$$\mathsf{AND}(f) = O(\log^{5/c} \mathsf{Mon}(f) \cdot \log^{c\alpha} \mathsf{Mon}(f))$$

as proposed by the log-rank conjecture.

## 3 Condensing Monomial Sparsity

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

**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 $O(k^2)$ 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 2 is only relevant in the particular case of low monomial sparsity. The proof of Theorem 2 follows from a more general combinatorial lemma about set systems.

## 3.1 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 2** *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 $E_i \not\subseteq E_j$. $\square$

## 3.2 Proof of Theorem 2

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 $Q$.

## 3.3 Sharpness

At the time of writing, the question remains open as to whether the $O(k^2)$ condensation given in Theorem 2 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 $\prod\limits_{j \neq i} x_j$. 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 $M_1 = \prod\limits_{j \neq i} x_i$ remains unchanged, while the monomial $M_2 = \prod\limits_{j=1}^{n} x_j$ 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

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

## 4 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.

### 4.1 Condensing With Respect to Monomial Sparsity

One result for condensation of hitting set size comes as a corollary of Theorem 2:

**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 2, 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 $S_2$.

## 4.2 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.

# 5 Condensing AND Decision Tree Depth

## 5.1 The Topology of AND Decision Trees

As noted previously, the condensation for monomial sparsity constructed in the proof of Theorem 2 preserves quite a lot of the structure of the polynomial, beyond just the number of monomials. The construction gives a bijection between the monomials $M_1 ... M_k$ of $f$ and the monomials $M_1' ... M_k'$ of the condensation $g$ that is a set system isomorphism, and corresponding monomials $M_i$ and $M_i'$ have the same coefficients. Given this similarity between the polynomial representations of $f$ and $g$, it is reasonable to wonder whether the optimal AND decision trees for $f$ and $g$ also exhibit some sort of similarity, such as being isomorphic as graphs.

More weekly, to obtain the desired condensation result for $\mathsf{AND}$, it would suffice to show that $\mathsf{AND}(f)$ is polynomially bounded above by $\mathsf{AND}(g)$. One route to such a result would be to somehow use an AND decision tree for $g$ to construct an AND decision tree for $f$. At the time of writing, the existence of such a condensation result for $\mathsf{AND}$ has not yet been established. However, the following lemma establishes a connection between the monomial sets of a boolean function $f$ and the node sets in an AND decision tree for $f$, suggesting this approach may hold promise as a future line of inquiry.

**Lemma 3** *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function with monomials $\{M_1 ... M_k\}$. Let $\mathcal{T}$ be the topology on $[n]$ generated by taking $\{M_1 ... M_k\}$ as a sub-basis. There exists an optimal AND decision tree $T$ for $f$ such that every node $N$ of $T$ is open in $\mathcal{T}$.*

By the definition of a topology generated by a sub-basis, $\mathcal{T}$ has a basis $\mathcal{B}$ of all finite intersections of $\{M_1...M_k\}$. The space $[n]$ is finite, so the topology $\mathcal{T}$ can have at most $2^{|n|}$ distinct open sets, since that is the size of the discrete topology. Because $\mathcal{T}$ is finite, the notion of a minimal open superset of a set is well-defined, which is not the case in general topological spaces. In particular, each $i \in [n]$ has a minimal open neighborhood $N_i$. $N_i$ is necessarily a basis element, since for any topological space $X$, if $x \in U \subseteq X$ where $U$ is open, then there exists a basis element $B$ such that $x \in B \subseteq U$. By the construction of the basis $\mathcal{B}$, we have

$$N_i = \bigcap_{\{M_r : i \in M_r\}} M_r$$

What we will show is that given any AND decision tree $T$ for $f$, we create an AND decision tree $T'$ that still correctly decides $f$, in which each node $N$ of $T$ has been replaced by an open $N'$ with $N \subseteq N'$. In particular, this $N'$ will be given by

$$N' = \bigcup_{i \in N} N_i$$

To show that $T'$ correctly decides $f$, it suffices by induction to instead prove the correctness of a tree $T_1$, in which, for a single $i \in [n]$, each node $N$ with $i \in N$ is replaced by $N \cup N_i$. Additionally, if $i \neq j \neq k \in [n]$, and $j \in N_i$ and $k \in N_j$, then $k \in N_i$. This is because $k \in N_j$ iff every open $U \subseteq [n]$ with $j \in U$ also has $k \in U$. If $j \in N_i$, then $N_i$ is an open set containing $j$, so $k \in N_i$ as well. Thus, the correctness of $T_1$ again follows inductively from the correctness of a tree $T_{ij}$ in which, for some $i$ with $j \neq i \in N_i$,[1] each node $N$ with $i \in N$ has been replaced by $N \cup \{j\}$.

To show that $T_{ij}$ correctly decides $f$, choose some $x \in \{0,1\}^n$. When $x_j = 1$, $x$ will have the same evaluation path in $T_{ij}$ and $T$. This is because when $x_j = 1$, $N(x) = (N \cup \{j\})(x)$, where the sets $N$ and $N \cup \{j\}$ are identified with their respective monomials for convenience of notation. When $x_j = 0$, $(N \cup \{j\})(x) = 0$ for any monomial $N$. The tree $T_{ij}$ replaces the node $N$ in $T$ by $N \cup \{j\}$ only when $i \in N$. If $x_i = 0$, then $N(x) = 0$ already, so $x$ again has the same evaluation path in $T_{ij}$ as in $T$. What remains is the case where $x_i = 1$ but $x_j = 0$. Let $y \in \{0,1\}^n$ be identical to $x$ except that $y_i = 0$. The evaluation path for $x$ in $T_{ij}$ is the same as the path for $y$ in $T$. If $f(x) = f(y)$, then $x$ is still correctly classified in $T_{ij}$, and it turns out that this is, in fact, the case. To see this, consider any monomial $M_r \in \{M_1...M_k\}$. If $i \in M_r$, then since $j \in N_i$, $j \in M_r$ as well, and $x_j = y_j = 0$, so $M_r(x) = M_r(y) = 0$. If $i \notin M_r$, then $j \notin M_r$ either, so because $x_k = y_k$ for every $k \neq j$, we again have $M_r(x) = M_r(y)$. Since $M_r(x) = M_r(y)$ for every monomial $M_r$ of $f$, $f(x) = f(y)$. $\square$

## 5.2 Gluing Indistinguishable Points is Insufficient

Suppose $i, j \in [n]$ are indistinguishable in the topology $\mathcal{T}$ generated by the sub-basis of the monomials $\{M_1...M_k\}$ of $f$ in that any open set $E$ containing either $i$ or $j$ contains both $i$ and $j$. Then every monomial $M_r$ containing either $i$ or $j$ has $i, j \in M_r$. In terms of the minimal neighborhoods defined in the proof of Lemma 3, this is equivalent to having $j \in N_i$ and $i \in N_j$. Note that this implies that the topology $\mathcal{T}$ does not satisfy the $T_0$ separation axiom.[2] If $x, y \in \{0,1\}^n$ are such that $x_k = y_k \forall k \neq i, j$ and $y_i = y_j = x_i \cdot x_j$, then $f(x) = f(y)$, since $M_r(x) = M_r(y)$ for every monomial of $f$, just as in the proof of Lemma 3.

Define an equivalence relation $\sim$ on $[n]$ by $i \sim j$ iff $i, j$ are topologically indistinguishable in the sense described above. To verify that $\sim$ does in fact define an equivalence relation, recall that if $j \in N_i$ and $k \in N_j$, then $k \in N_i$. If $i \sim j$ and $j \sim k$, we have $j \in N_i$ and $k \in N_j$, implying $k \in N_i$, as well as $i \in N_j$ and $j \in N_k$, implying $i \in N_k$. Let $C_1...C_m$ denote the equivalence classes of $\sim$. If $i \sim j$, then for any $x$, $x_i, x_j$ can both be replaced by $x_i \cdot x_j$ without affecting the value of $f$. Inductively, if all $x_i, i \in C_k$ are replaced by $\prod_{j \in C_k} x_j$, the value of $f$ at the resulting point will be the same as $f(x)$. Similarly, as shown in the proof of Lemma 3, if each node $N$ in a decision tree $T$ for $f$ containing a member of $C_k$ is replaced by $N \cup C_k$, the resulting tree will still decide $f$.

Let $A \subseteq [n]$ contain one representative $a_j$ from each equivalence class $C_j$. Define $g : \{0,1\}^m \to \{0,1\}$ to be the subfunction of $f$ obtained by fixing $x_i$ to 1 for all $i \notin A$. Let $T$ be any AND decision tree for $g$, and

[1] When $N_i$ is a singleton, replacing each $N$ having $i \in N$ with $N \cup N_i$ does not alter the tree at all.

[2] A topology $\mathcal{T}$ on a space $X$ is $T_0$ if for every pair $x \neq y \in X$, there exists an open set $U$ containing exactly one of $x, y$.

define $T'$ to be the AND decision tree obtained by, for $j \in 1...m$, replacing each occurrence of $a_j$ in $T$ by $C_j$. Then $T'$ is an AND decision tree that decides $f$. Since $g$ is a subfunction of $f$, $\mathsf{AND}(g) \leq \mathsf{AND}(f)$, so we have $\mathsf{AND}(g) = \mathsf{AND}(f)$. If $\mathsf{Mon}(f) = k$ and the relation $\sim$ for the topology generated by the monomials of $f$ has $m$ equivalence classes, then by Theorem 1,

$$\mathsf{AND}(f) = O(\log^5(k)\log(m))$$

In particular, to obtain a condensation theorem that resolves log-rank for AND functions, it would suffice to prove the claim that if $f$ has $k$ monomials, then the number of equivalence classes of $\sim$ is $O(2^{\log^c(k)}$ for some constant $c$.

Unfortunately, this claim is not true. **TODO: example**

## References

[Göö+24] Mika Göös 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/.

[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 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.

[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.

[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.

[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.