

# On the Complexity of Hazard-Free Formulas

LEAH LONDON ARAZI\*, Tel Aviv University, Israel

AMIR SHPILKA\*, Tel Aviv University, Israel

This paper studies the *hazard-free formula complexity* of Boolean functions.

Our first result shows thatunate functions are the only Boolean functions for which the monotone formula complexity of the hazard-derivative equals the hazard-free formula complexity of the function itself. Consequently, they are the only functions for which the hazard-derivative approach of Ikenmeyer et al. (J. ACM, 2019) yields optimal bounds.

Our second result proves that the hazard-free formula complexity of a uniformly random Boolean function is at most  $2^{(1+o(1))n}$ . Prior to this, no better upper bound than  $O(3^n)$  was known. Notably, unlike in the general case of Boolean circuits and formulas, where the typical complexity is derived from that of the multiplexer function with  $n$ -bit selector, the hazard-free formula complexity of a random function is smaller than the optimal hazard-free formula for the multiplexer by an exponential factor in  $n$ .

We provide two proofs of this fact. The first is direct, bounding the number of prime implicants of a random Boolean function and using this bound to construct a DNF of the claimed size. The second introduces a new and independently interesting result: a weak converse to the hazard-derivative lower bound method, which gives an upper bound on the hazard-free complexity of a function in terms of the monotone complexity of a subset of its hazard-derivatives.

Additionally, we explore the hazard-free formula complexity of block composition of Boolean functions and obtain a result in the hazard-free setting that is analogous to a result of Karchmer, Raz, and Wigderson (Computational Complexity, 1995) in the monotone setting. We show that our result implies a stronger lower bound on the hazard-free formula depth of the block composition of the set covering function with the multiplexer function than the bound obtained via the hazard-derivative method.

Additional Key Words and Phrases: Hazard-free computation, Boolean formulas, monotone formulas, Karchmer-Wigderson games, communication complexity, lower bounds

## 1 Introduction

*Hazards* are a critical issue in the design of combinational circuits, which are key components in digital systems, realizing Boolean functions in the physical domain. A combinational circuit consists of gates connected by electrical wires. When setting voltage at the inputs, the signal propagates through the gates and wires until it reaches the output gate. This propagation involves inevitable signal *delays*, which may vary on each path of the circuit. As a result, when the inputs change, some gates may respond faster than others, potentially leading to *static hazards*. A static hazard occurs when a change in the input causes a brief, unintended fluctuation in the output, even though the output is expected to remain constant. Such hazards in combinational circuits can lead to unpredictable system behavior.

The study of hazards began in the late 1950s with the pioneering work of [11], who suggested that it is possible to design circuits whose structure inherently prevents hazards without introducing delays. Huffman was the first to provide a hazard-free formula implementation for all Boolean functions, achieving a size of  $O(\sqrt{n} \cdot 3^n)$ . Later, additional works classified different types of hazards and suggested methods for identifying and eliminating them [23, 37]. There is a broad body of

---

\*This research was co-funded by the European Union (ERC, EACTP, 101142020), the Israel Science Foundation (grant number 514/20) and the Len Blavatnik and the Blavatnik Family Foundation. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Table 1. Extended De Morgan gates.

or	0	u	1	and	0	u	1	not	0	u	1
0	0	u	1	0	0	0	0		1	u	0
u	u	u	1	u	0	u	u				
1	1	1	1	1	0	u	1				

literature on hazard-free computation, and we refer the interested reader to [12, 24] and references therein.

Recent research on hazard-free computation focuses on two main areas: electronic circuit design (e.g., [3, 7]) and computational complexity (e.g., [12–14, 20]). While the motivation for the former is clear, the interest from the computational complexity community stems from the results of [12], which show that hazard-free formula complexity generalizes monotone formula complexity to all Boolean functions. This makes it an interesting restricted computational model of Boolean circuits and raises hope that hazard-free techniques may offer deeper insights into the general model.

Kleene’s three-valued logic was introduced as an abstraction of static hazards [6, 38]. This logic extends the standard Boolean values  $\{0, 1\}$  by introducing a third symbol,  $u$ , representing an undefined or unstable value.

**Definition 1.1** (Kleene three-valued logic [19]). *Kleene’s three-valued strong logic of indeterminacy extends the two-valued Boolean logic by a third value  $u$ . The Boolean values  $\{0, 1\}$  are called “stable” and  $u$  is called “unstable”. The De Morgan gates are appropriately extended, as shown in Table 1.*

Now, any Boolean circuit or formula can be viewed as taking an input  $x \in \{0, u, 1\}^n$  and outputting a value in  $\{0, u, 1\}$ . The extended gates are monotone with respect to the following partial order:

**Definition 1.2** ( $\stackrel{u}{\leq}$  [26]). *The relation  $\stackrel{u}{\leq}$  is a partial order in which  $u \stackrel{u}{\leq} 1$ ,  $u \stackrel{u}{\leq} 0$  and  $0, 1$  are not comparable. For  $x, y \in \{0, u, 1\}^n$ , we denote  $x \stackrel{u}{\leq} y$  if for every  $i \in [n]$ ,  $x_i \stackrel{u}{\leq} y_i$ .*

We present an abstraction of hazards using ternary algebra, following the definitions of [7, 12].

**Definition 1.3** (Resolution). *Let  $x \in \{0, u, 1\}^n$ . A resolution  $y \in \{0, 1\}^n$  of  $x$  is obtained by replacing every occurrence of  $u$  in  $x$  by either 0 or 1. We denote by  $R(x)$  the set of all resolutions of  $x$ , namely,  $R(x) := \{y \in \{0, 1\}^n \mid x \stackrel{u}{\leq} y\}$ .*

**Definition 1.4** (Hazard). *Let  $C$  be a Boolean circuit.  $C$  has a hazard at input  $x \in \{0, u, 1\}^n$  if and only if  $C(x) = u$  and there exists  $b \in \{0, 1\}$  such that for every  $y \in R(x)$ , it holds that  $C(y) = b$ .*

In words, there is a hazard at input  $x \in \{0, u, 1\}^n$  if the function is constant on every resolution of  $x$  (i.e., regardless of how the unstable coordinates are resolved into stable values, the output remains unchanged), yet the circuit still outputs  $u$ . Put differently, a hazard-free circuit outputs  $u$  if and only if the output cannot be determined solely by the stable bits.

**Definition 1.5** (Hazard-free circuit). *Let  $C$  be a Boolean circuit.  $C$  is said to be hazard-free if for every  $x \in \{0, u, 1\}^n$ ,  $C$  does not have a hazard at  $x$ .*

By examining the truth tables in Table 1, we observe that the extended De Morgan gates are hazard-free. However, circuits<sup>1</sup> and formulas composed of these gates do not necessarily retain the hazard-free property. Consider  $F(x) = x \vee \neg x$ , a Boolean formula over one bit.  $F$  computes the

<sup>1</sup>An example of a Boolean circuit exhibiting hazards is shown in [12, Figure 1].

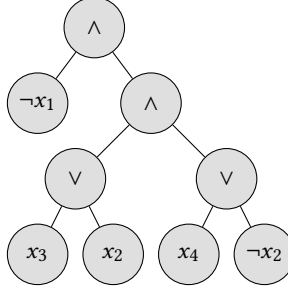


Fig. 1. Formula with a hazard at  $x_1 = u, x_2 = u, x_3 = 0, x_4 = 0$ .

constant Boolean function 1, and by the truth tables in Table 1, has a hazard at  $x = u$ . Another less trivial example of a Boolean formula exhibiting hazards, is the formula in Figure 1. It is not difficult to verify that when evaluated on every resolution in  $R(uu00)$  the formula outputs 0, but on  $uu00$ , the formula outputs  $u$ .

Not every ternary function  $f : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  can be computed by a Boolean circuit. For instance, consider  $f$  over one bit, defined such that  $f(0) = f(1) = u$  and  $f(u) = 1$ . The reason  $f$  cannot be computed by a circuit is that every gate in Table 1 outputs a Boolean value when given a Boolean input and, as mentioned, is monotone with respect to  $\leq$ . It can be proven by induction that circuits also preserve these properties. Consequently, only a subset of all ternary functions can be computed by Boolean circuits. [26] refers to these functions as *natural functions*.

**Definition 1.6** (Natural function). *A function  $f : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  is called a natural function if it satisfies the following:*

- (1)  *$f$  preserves stable values, i.e., for every Boolean input  $x \in \{0, 1\}^n$  we have  $f(x) \in \{0, 1\}$ .*
- (2)  *$f$  is monotone with respect to  $\leq$ . Intuitively, replacing stable bits in the input with  $u$  can only cause the output to change to  $u$ .*

**Proposition 1.7** ([26, Theorem 3]). *A function  $f : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  can be computed by a Boolean circuit if and only if  $f$  is natural.*

An equivalent way of defining a hazard-free circuit is by the ternary function it computes. A circuit is considered hazard-free if it calculates the following natural function:

**Definition 1.8** (The hazard-free extension of  $f$ ). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The hazard-free extension of  $f$ , denoted  $\tilde{f} : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$ , is defined as follows:*

$$\tilde{f}(x) = \begin{cases} 0, & \text{if } f(y) = 0 \text{ for all } y \in R(x), \\ 1, & \text{if } f(y) = 1 \text{ for all } y \in R(x), \\ u, & \text{otherwise.} \end{cases}$$

[12] showed a profound correspondence between monotone complexity and hazard-free complexity. Recall that a Boolean function is monotone if changing the value of an input coordinate from 0 to 1 cannot change the value of the function from 1 to 0. Monotone functions can be computed by monotone circuits, which are circuits that do not use negation gates. Using the hazard-derivatives method, which we discuss next, they proved that for monotone functions, the hazard-free complexity and the monotone complexity are identical.

**THEOREM 1.9 (HAZARD-FREE AND MONOTONE CIRCUIT COMPLEXITY ARE EQUIVALENT [12, THEOREM 1.3]).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function, then:*

$$\text{size}_C^u(f) = \text{size}_C^+(f),$$

where  $\text{size}_C^+(f)$  and  $\text{size}_C^u(f)$  denote the minimal size of a monotone Boolean circuit and a hazard-free circuit computing  $f$ , respectively.

In fact, [14] further proved that an optimal hazard-free circuit for a monotone function must be a monotone circuit. As superpolynomial lower bounds for monotone circuits [1, 30, 36] are known, [12] used Theorem 1.9 to prove lower bounds on the hazard-free circuit complexity of monotone functions whose Boolean circuit complexity is only polynomial.

While Theorem 1.9 only concerns monotone functions, [12] found a striking connection between hazard-free complexity and monotone complexity for all Boolean functions. That is, the hazard-free complexity of every Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is bounded from below by the monotone complexity of the *hazard-derivative* of  $f$ , defined as follows:

**Definition 1.10** (The hazard-derivative of a natural function [12, Definition 4.3]). *Let  $f \in \{0, u, 1\}^n$  be a natural ternary function. The hazard-derivative of  $f$ , denoted  $df : \{0, 1\}^{2n} \rightarrow \{0, 1\}$ , is defined as follows:*

$$\forall x, y \in \{0, 1\}^n, \quad df(x; y) = \begin{cases} 0, & \text{if } f(x + uy) = f(x), \\ 1, & \text{if } f(x + uy) = u. \end{cases}$$

Where for every  $i \in [n]$ :

$$(x + uy)_i = \begin{cases} x_i, & \text{if } y_i = 0, \\ u, & \text{if } y_i = 1. \end{cases}$$

For a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , we define  $df := d\tilde{f}$ .

The  $+$  operator in Definition 1.10 also serves as the hazard-free XOR operator (see Definition 2.1).<sup>2</sup>

For a fixed  $x \in \{0, 1\}^n$ ,  $df(x; y)$  indicates whether “perturbing”  $x$  according to  $uy$  causes  $\tilde{f}$  to output an unstable value. It is not difficult to see that  $df(x; \cdot)$  is a monotone Boolean function, as if we perturb more bits in  $x$ , we are more likely to have two resolutions on which  $f$  outputs different values.

**Lemma 1.11** (The hazard-derivative is monotone [12, Lemma 4.6]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function and  $x \in \{0, 1\}^n$  be a fixed input, then  $df(x; \cdot)$  is a monotone Boolean function.*

Every hazard-free circuit (or formula) for  $f$  can be tweaked to yield a monotone circuit (or formula) for  $df(x; \cdot)$  resulting in the following Theorem:<sup>3</sup>

**THEOREM 1.12 (THE HAZARD-DERIVATIVE LOWER BOUND [12, THEOREM 4.9]).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function and  $x \in \{0, 1\}^n$  be a fixed Boolean input, then:*

$$\text{size}_C^+(df(x; \cdot)) \leq \text{size}_C^u(f) \quad \text{and} \quad \text{size}_F^+(df(x; \cdot)) \leq \text{size}_F^u(f),$$

where  $\text{size}_F^+(f)$  and  $\text{size}_F^u(f)$  denote the minimal size of a monotone Boolean formula and a hazard-free formula computing  $f$ , respectively.

<sup>2</sup>When considering  $uy_i = \begin{cases} 0, & \text{if } y_i = 0, \\ u, & \text{if } y_i = 1. \end{cases}$

<sup>3</sup>[12, Theorem 4.9] only speaks about circuit size, but the same proof holds for formulas as well.

[12], and later [14], provided examples of non-monotone Boolean functions that have polynomial-sized Boolean circuits but whose hazard-derivatives require large monotone circuit complexity, thereby establishing strong lower bounds on the hazard-free circuit complexity of these functions. We note that until [12], hazard-free lower bounds were proved only for restricted models such as hazard-free depth 2 circuits [6, 28]. Therefore, their work motivates the study of the hazard-free model as a bridge between monotone complexity and Boolean circuit complexity.

Additionally, [12, Theorem 1.3] proved that the hazard-derivative method yields *tight* bounds for monotone Boolean functions. That is, for a monotone function  $\text{size}_C^+(df(x; \cdot)) = \text{size}_C^u(f)$  and  $\text{size}_F^+(df(x; \cdot)) = \text{size}_F^u(f)$  when  $x = \bar{0}$ . This raises the following intriguing question.

**Question 1.13.** *Is there a criterion that determines whether the hazard-derivative method yields an exact lower bound? If so, is this criterion easy to verify?*

### 1.1 The monotone gap

[12, 14] obtained exponential lower bounds on the hazard-free computation of many Boolean functions using Theorem 1.12. However, there are Boolean functions for which the hazard-derivative method fails to provide optimal lower bounds on hazard-free complexity due to the low monotone complexity of the hazard-derivatives. [13] highlights this issue and refers to it as the *monotone barrier*. They define breaking the monotone barrier as achieving stronger lower bounds than those obtained by the hazard-derivative method.

As an example of a result that breaks the monotone barrier, [13] note that the hazard-derivatives of the parity function on  $n$  variables,  $\text{XOR}_n$ , are simply OR functions on  $n$  literals, which have trivial formulas of size  $n$ . On the other hand, it is known that the De Morgan formula complexity of  $\text{XOR}_n$  is  $\Omega(n^2)$ , while it is not hard to see that any formula for  $\text{XOR}_n$  is hazard-free.

To quantify the gap between the hazard-free complexity of a given Boolean function and the monotone complexity of its hazard-derivatives, we define a measure, similar the one defined in [14, Section 7], which we refer to as the *monotone gap*.

**Definition 1.14** (The monotone gap). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The monotone gap of  $f$  is defined as:*

$$\text{mono-gap}(f) := \frac{\text{size}_F^u(f)}{\max_{x \in \{0, 1\}^n} \text{size}_F^+(df(x; \cdot))}.$$

For example, the monotone gap of  $\text{XOR}_n$  is  $\Omega(n)$ . As another example, in Section 4.1, we prove that the monotone gap of Andreev's function [2] is  $\Omega\left(\frac{n^2}{\log n}\right)$ . In an attempt to understand the power of the hazard-derivative method (or its weaknesses), we wish to answer the following:

**Question 1.15.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. How large can  $\text{mono-gap}(f)$  be? What is  $\text{mono-gap}(f)$  for a typical Boolean function?*

### 1.2 The hazard-free KW game

[13] studied a generalization of the classic Karchmer-Wigderson game [18] in the hazard-free setting (see Section 2.2 for background on KW games).

**Definition 1.16** (Hazard-free KW game [13, Definition 5]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The hazard-free Karchmer-Wigderson game of  $f$ , denoted  $\text{KW}_f^u$ , is defined as follows: Alice receives  $x \in \{0, u, 1\}^n$  such that  $\tilde{f}(x) = 1$ , Bob receives  $y \in \{0, u, 1\}^n$  such that  $\tilde{f}(y) = 0$ , and their goal is to output a coordinate  $i \in [n]$  such that  $x_i \neq y_i$  and  $x_i, y_i \neq u$ .*

As  $KW_f^u \subseteq \tilde{f}^{-1}(1) \times \tilde{f}^{-1}(0) \times [n]$ , and every  $x \in \tilde{f}^{-1}(1)$  and  $y \in \tilde{f}^{-1}(0)$  are valid inputs<sup>4</sup>,  $KW_f^u$  has a potentially larger set of inputs than the classic game. As a result, the communication matrix may be more complex. Similarly to the classic KW game, the hazard-free KW game preserves a tight connection between the underlying communication problem complexity and the optimal hazard-free formula depth and size.

**THEOREM 1.17** ([13, THEOREM 7]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then,*

$$\text{depth}_F^u(f) = CC(KW_f^u) \text{ and } \text{size}_F^u(f) = \text{mono-rec}(KW_f^u),$$

where  $\text{depth}_F^u(f)$  is the depth of the shallowest hazard-free formula computing  $f$ ,  $CC(\mathcal{R})$  is the communication complexity of the relation  $\mathcal{R}$ , and  $\text{mono-rec}(\mathcal{R})$  is the minimal number of leaves in a protocol that solves  $\mathcal{R}$ .

Like the monotone version of the KW game, [13] proved that the communication complexity of  $KW_f^u$  does not decrease even if it is played only on the *prime implicants* of  $f$  and the *prime implicants* of  $f$ .

**THEOREM 1.18** ([13, THEOREM 10]). *For any Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , the communication complexity of  $KW_f^u$  remains unchanged even if we restrict Alice's input to the prime implicants of  $f$  and Bob's input to the prime implicants of  $f$ .*

The notion of prime implicants and prime implicants is well established in Boolean algebra. Following [13], we provide analogous definitions using ternary algebra as previously defined in [20][p.3] and [13][p.7].

**Definition 1.19** (Implicants and implicants of a Boolean function). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. We define the implicants of  $f$ , denoted by  $I_1^{(f)}$ , and the implicants of  $f$ , denoted by  $I_0^{(f)}$ , as follows:*

$$I_1^{(f)} := \tilde{f}^{-1}(1), \quad I_0^{(f)} := \tilde{f}^{-1}(0).$$

**Definition 1.20** (Prime implicants and prime implicants of a Boolean function). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. We say  $x \in \{0, u, 1\}^n$  is a prime implicant of  $f$  if the following holds:*

- $x$  is an implicant of  $f$ .
- For every stable bit  $x_i \in \{0, 1\}$  in  $x$ , we have  $\tilde{f}(x|_{i \leftarrow u}) = u$ , where  $x|_{i \leftarrow b}$  stands for replacing the value of the  $i$ 'th coordinate in  $x$  with  $b \in \{0, u, 1\}$ .

A prime implicate is defined analogously. We denote the set of prime implicants of  $f$  by  $P_1^{(f)}$  and the set of prime implicants of  $f$  by  $P_0^{(f)}$ .

For the remainder of the paper, we abuse notation and use  $KW_f^u$  to denote the restricted game. We next define the communication matrix of the  $KW_f^u$  game.

**Definition 1.21** (Communication matrix of  $KW_f^u$ ). *Let  $\mathcal{X} := (p_1^{(1)}, \dots, p_1^{(k)})$  and  $\mathcal{Y} := (p_0^{(1)}, \dots, p_0^{(\ell)})$  be an arbitrary ordering of the elements in  $P_1^{(f)}$  and  $P_0^{(f)}$ , respectively. We call  $\mathcal{X}$  and  $\mathcal{Y}$  the row labels and column labels, respectively. The communication matrix of  $KW_f^u$ , denoted  $M_{KW_f^u}$ , is defined such that every entry of  $M_{KW_f^u}$  contains the set of all stable coordinates in which  $p_1^{(i)}$  and  $p_0^{(j)}$  differ:*

$$\forall i \in [k], j \in [\ell], \quad (M_{KW_f^u})_{i,j} = \{z \in [n] : (p_1^{(i)} + p_0^{(j)})_z = 1\}.$$

<sup>4</sup>There must exist a coordinate  $i \in [n]$  such that  $x_i \neq y_i$  and  $x_i, y_i$  are stable. Otherwise,  $x, y$  have a common resolution, in contradiction to Definition 1.8.

For  $y \in \{0, u, 1\}^n$ , we denote by  $y|_{s \leftarrow a}$  the result of replacing all occurrences of  $s \in \{0, u, 1\}$  in  $y$  with  $a \in \{0, u, 1\}$ . We then use  $(p_1^{(i)} + p_0^{(j)})|_{u \leftarrow 0} := \{z \in [n] : (p_1^{(i)} + p_0^{(j)})_z = 1\}$  as an abbreviated notation.

Another interesting observation is that, for a monotone function, the hazard-free KW game captures the monotone formula complexity.

**THEOREM 1.22** ([13, THEOREM 11]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Then,  $KW_f^u$  and  $KW_f^+$  are equivalent.*

Consequently, the size and depth of an optimal hazard-free formula are identical to those of its monotone counterpart. This provides a proof of Theorem 1.9 for formulas instead of circuits.

By using communication complexity methods to prove lower bounds on hazard-free formulas, [13, Theorems 19, 23] determined the exact hazard-free formula size of the multiplexer function.

**Definition 1.23** (The multiplexer function [13]). *The function  $MUX_n : \{0, 1\}^{n+2^n} \rightarrow \{0, 1\}$ , called the multiplexer function, is defined by*

$$MUX_n(s, x) = x_{\text{bin}(s)},$$

where  $s \in \{0, 1\}^n$ ,  $x \in \{0, 1\}^{2^n}$  and  $\text{bin}(s)$  is the natural number represented by  $s$  in binary basis.

**THEOREM 1.24** (HAZARD-FREE FORMULA SIZE FOR THE MULTIPLEXER FUNCTION [13, EXACT BOUNDS]).

$$\text{size}_F^u(MUX_n) = 2 \cdot 3^n - 1.$$

This result is much better than what can be achieved using the hazard-derivative method since the hazard-derivatives of the multiplexer function have quasi-linear monotone formula complexity (see [13, Proposition 1]).

### 1.3 Universal upper bounds

An immediate consequence of Theorem 1.24 is that any Boolean function  $f$  can be implemented by a hazard-free formula of size  $2 \cdot 3^n - 1$ . Indeed, it holds that:

$$f(x) = MUX_n(x, f(0, 0, \dots, 0), \dots, f(1, 1, \dots, 1)),$$

therefore, the hazard-free formula for the multiplexer, when hardwired the truth table of  $f$ , gives a hazard-free formula for  $f$ . This result is the first to break the  $O(\sqrt{n} \cdot 3^n)$  upper bound of Huffman [11].

We note that this state of knowledge is much poorer compared to what is known for hazard-free circuits. Analogous to [27, 34], Jukna [14, Section 7] proved that any Boolean function has a hazard-free circuit of size  $O(2^n/n)$ , applying the following:

**Proposition 1.25** ([14, Proposition 3]). *Let  $C_0, C_1$  be arbitrary hazard-free circuits over  $x_1, \dots, x_{n-1}$ , and  $x_n$  a new variable. Then, the circuit*

$$C := (\neg x_n \wedge C_0) \vee (x_n \wedge C_1) \vee (C_0 \wedge C_1) \quad (1)$$

*is hazard-free.*

Therefore, naturally, we can construct a hazard-free circuit for  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  by using hazard-free implementations  $C_0$  and  $C_1$  for  $f_0(x_1, \dots, x_{n-1}) := f(x_1, \dots, x_{n-1}, 0)$  and  $f_1(x_1, \dots, x_{n-1}) := f(x_1, \dots, x_{n-1}, 1)$ , respectively. However, in the case of formulas, the size of the formula obtained by (1) is  $O(3^n)$ , since it is composed of 3 hazard-free formulas over  $n - 1$  bits.<sup>5</sup>

<sup>5</sup>Note that the extra  $(C_0 \wedge C_1)$  term is necessary. Say that for some  $x \in \{0, u, 1\}^{n-1}$  we have  $\tilde{f}(x, u) = 1$ . Hence, by definition, we also have  $\tilde{f}_0(x) = 1, \tilde{f}_1(x) = 1$ , and so  $C_0(x) = 1, C_1(x) = 1$ . But,  $(\neg x_n \wedge C_0) \vee (x_n \wedge C_1)$  will evaluate to  $u$ .

This result matches the lower bound obtained via the same counting arguments used in the Boolean setting. In particular, for a typical Boolean function (i.e., one sampled uniformly at random from all Boolean functions on  $n$  bits), we have  $\text{size}_C^u(f) = \Theta(\text{size}_C(f))$ .

On the other hand, counting arguments give a lower bound of order  $\Omega(2^n/\log n)$  for the size of the hazard-free formula of a random Boolean function [33, 34], whereas Theorem 1.24 only implies an  $O(3^n)$  upper bound. Thus, there is an exponential gap between the two bounds. This naturally leads to the following question.

**Question 1.26.** Denote with  $\text{size}_F^u(n)$  the maximal hazard-free complexity of an  $n$ -bit Boolean function. What is the asymptotic value of  $\text{size}_F^u(n)$ ? What is  $\text{size}_F^u(f)$  of a typical Boolean function?

#### 1.4 The hazard-free KRW conjecture

Karchmer, Raz and Wigderson [17] introduced the KRW conjecture as an approach for establishing super-polynomial lower bounds for De Morgan formulas. Proving this conjecture would imply  $P \not\subseteq \text{NC}^1$ . A detailed discussion can be found in [25].

To state the conjecture, we define the block-composition of two Boolean functions  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  as

$$f \diamond g(X) := f(g(X_1), \dots, g(X_n)),$$

where  $X$  is a  $n \times m$  Boolean matrix, and  $X_i \in \{0, 1\}^m$  is the  $i$ 'th row of  $X$ . For a Boolean function  $f$ , we denote with  $\text{depth}_F(f)$  the depth of the shallowest formula computing  $f$ .

**Conjecture 1.27** (The KRW conjecture [17]). Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be non-constant Boolean functions. Then it holds that:<sup>6</sup>

$$\text{depth}_F(f \diamond g) \approx \text{depth}_F(f) + \text{depth}_F(g).$$

Following Theorem 2.14, the conjecture can be rephrased as:

$$\text{CC}(KW_{f \diamond g}) \approx \text{CC}(KW_f) + \text{CC}(KW_g).$$

Roughly, the conjecture states that the smallest depth formula for computing  $f \diamond g$  is obtained from the composition of the shallowest formula for  $f$  and the shallowest formula for  $g$ .

We next state an analogous conjecture in the hazard-free setting.

**Conjecture 1.28** (Hazard-Free KRW conjecture). Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be non-constant Boolean functions. Then the following holds:

$$\text{depth}_F^u(f \diamond g) \approx \text{depth}_F^u(f) + \text{depth}_F^u(g).$$

Similarly to [4, 32], we state a stronger version of the conjecture on the formula size.

**Conjecture 1.29** (Hazard-free strong KRW conjecture). Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be non-constant Boolean functions. Then the following holds:

$$\text{size}_F^u(f \diamond g) \approx \text{size}_F^u(f) \cdot \text{size}_F^u(g).$$

The KRW conjecture has been extensively studied through various special cases and simplified communication problems, as intermediate steps toward proving Conjecture 1.27. The monotone version of the KRW conjecture is most related to the hazard-free setting due to the connection to the hazard-derivative, which is a monotone function.

<sup>6</sup>We use  $\approx$  instead of  $\geq$  as we can allow a less exact inequality, see [8, 17].



**Conjecture 1.30** (Monotone KRW conjecture [17]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be monotone Boolean functions. Then the following holds:*

$$\text{depth}_F^+(f \diamond g) \approx \text{depth}_F^+(f) + \text{depth}_F^+(g).$$

In [32, Theorem 3.1], Conjecture 1.30 was proved for monotone inner functions  $g$  whose depth complexity can be lower-bounded by a lifting theorem. In an earlier work, [17] observed that the monotone KW game can be played on the prime implicants and implicates of the monotone function, aligning well with the results concerning the hazard-free KW game presented in Theorem 1.22. Notably, the hazard-free KRW conjecture generalizes the monotone KRW conjecture, as proving the former for every  $f$  and  $g$  would also imply the latter.

### 1.5 Our results

*Unateness as a criterion.* We provide a complete answer to Question 1.13 by proving that the hazard-derivative method yields exact lower bounds only for unate Boolean functions:

**THEOREM 1.31 (UNATENESS AS A CRITERION).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function. Then, there exists  $x \in \{0, 1\}^n$  such that:*

$$\text{size}_F^+(\text{df}(x; \cdot)) = \text{size}_F^u(f),$$

*if and only if  $f$  is a unate function.*

A unate Boolean function is defined as follows:

**Definition 1.32** (Unate function in  $x_i$ ). *A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is positive unate in  $x_i$  if for all  $x \in \{0, 1\}^n$  the following holds:*

$$f(x|_{i \leftarrow 0}) \leq f(x|_{i \leftarrow 1}),$$

*Similarly, a function is negative unate in  $x_i$  if for all  $x \in \{0, 1\}^n$  the following holds:*

$$f(x|_{i \leftarrow 0}) \geq f(x|_{i \leftarrow 1}).$$

*We say  $f$  is unate in  $x_i$  if  $f$  is either positive or negative unate in  $x_i$ .*

**Definition 1.33** (Unate function). *A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is unate if it is unate in  $x_1, \dots, x_n$ .*

Monotone functions are a subset of unate functions, which increase or decrease monotonically in each variable. Given the truth table of a Boolean function as input, we can efficiently determine whether it is unate or not by inspecting the truth table. We prove this result by analyzing the hazard-free KW game of the function and its hazard-derivatives.

*A universal formula upper bound for most Boolean functions.* We prove that the hazard-free formula size of a random Boolean function is  $2^{(1+o(1))n}$ , thereby partially answering Question 1.26. This gives an exponential improvement over the best known  $O(3^n)$  upper bound of [13]. It is important to note that the upper bound of [13] holds for all functions, so it is conceivable that some functions require hazard-free formula size larger than  $2^{(1+o(1))n}$ . Even if this is the case, then neither counting arguments nor the hazard-derivative method, which can only provide a lower bound of at most  $2^n$ , will be able to prove it.

**THEOREM 1.34 (A UNIVERSAL UPPER BOUND FOR MOST BOOLEAN FUNCTIONS).** *For a random Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , we get that with high probability:*

$$\frac{2^n}{\log n} \leq \text{size}_F^u(f) \leq 2^n \cdot n^{(1-o(1)) \log n}.$$

Thus, we demonstrate that, unlike Boolean circuits, Boolean formulas and hazard-free circuits – where the typical complexity matches that of the multiplexer function – the hazard-free formula complexity is smaller than the optimal hazard-free formula for the multiplexer function by an exponential factor in  $n$ .

We provide two proofs for Theorem 1.34 (see Sections 5.1 and 5.3). The second relies on two other results that are interesting on their own. The first shows that, with high probability, the hazard-derivatives of a random Boolean function do not have too large monotone formulas. In particular, this provides a strong answer to Question 1.15, showing that the monotone gap can be exponential in the number of variables.

**Proposition 1.35.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a random Boolean function. Then, with high probability,*

$$\text{mono-gap}(f) = \omega\left(\frac{2^n}{n^{\log n}}\right).$$

The second ingredient is an *upper bound* using hazard-derivatives. For  $b \in \{0, 1\}$ , we say that  $p_b \in P_b^{(f)}$  derives  $x$  if  $p_b \stackrel{u}{\leq} x$ . We denote the subset of  $P_b^{(f)}$  containing the strings that derive  $x$  by  $P_b^{(f)}|_x$ .

**THEOREM 1.36 (HAZARD-DERIVATIVES UPPER BOUND).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $b \in \{0, 1\}$ . Let  $X_b \subseteq f^{-1}(b)$ . Then,*

$$\text{if } P_b^{(f)} = \bigcup_{x \in X_b} P_b^{(f)}|_x, \text{ then it holds that } \text{size}_F^u(f) \leq \sum_{x \in X_b} \text{size}_F^+(df(x; \cdot)).$$

We also provide an example in Section 5.3.1 that shows that Theorem 1.36 yields tight bounds for range functions (functions that accept all inputs whose weight is within some range).

*Composition of hazard-free functions.* Following [17], we establish a lower bound on the hazard-free complexity of block-composition of functions and demonstrate how it can be used to surpass the hazard-derivative method.

**Proposition 1.37** (Lower bound via direct sum). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Let  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  a Boolean function. Let  $\Phi : X \times Y \rightarrow \{0, 1\}$  and  $\Psi : X' \times Y' \rightarrow \{0, 1\}$  be functions such that  $\Phi$  reduces to  $KW_f^*$  and  $\Psi$  reduces to  $KW_g^u$ . Then, the following holds:*

$$CC(KW_{f \diamond g}^u) \geq \log(\text{Rank}(M_\Phi)) + \log(\text{Rank}(M_\Psi)).$$

Proposition 1.37 provides a step toward advancing hazard-free lower bound techniques, and we hope it encourages further research in this direction. As an application, we prove a lower bound on the depth of the composition of the set covering function and the multiplexer function (see Section 6 for definitions).

**Proposition 1.38** (Composition of a set covering and the multiplexer function). *Let  $MUX_m$  be the multiplexer function and  $MC_{k,n}$  be the set covering function, such that  $k = c \log n$  for some suitable constant  $c > 0$ . Then,*

$$\text{depth}_F^u(MC_{k,n} \diamond MUX_m) \geq \log(3^m) + \log\left(\frac{n}{c \log n}\right) \geq \log(3) \cdot m + \Omega(c(\log n)^2).$$

## 1.6 Organization of the paper

In Section 2, we present some basic definitions, as well as basic properties of the implicants and implicates of Boolean functions, with a focus on monotone and unate functions. Additionally, we provide background on communication complexity and prove the simplification lemma (Lemma 2.12). In

Section 3, we study the KW hazard-free game of the hazard-derivative. In Section 4, we prove our first result (Theorem 1.31), and in Section 4.1, we present an example of an explicit function that exhibits a large monotone gap. Section 5 contains the proofs of the universal upper bound for most Boolean functions (Theorem 1.34), as well as the proof that a random Boolean function exhibits a large monotone gap (Proposition 1.35), and the hazard-derivatives upper bound result (Theorem 1.36). In Section 6, we study the composition of Boolean functions in the hazard-free setting and prove Proposition 1.37. We also demonstrate its applicability by proving Proposition 1.38. We conclude by presenting some open problems in Section 7.

## 2 Preliminaries

For an integer  $n$ , we set  $[n] := \{1, \dots, n\}$ . For  $x \in \{0, 1\}^n$ , we define  $\neg x$  to be the Boolean input such that for every  $i \in [n]$ , it holds that  $(\neg x)_i = \neg x_i$ . For  $x, y \in \{0, 1\}^n$ , we denote  $x \leq y$  if for every  $i \in [n]$ ,  $x_i \leq y_i$  (as integers).<sup>7</sup> A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is *monotone* if for every  $x, y \in \{0, 1\}^n$  such that  $x \leq y$  it holds that  $f(x) \leq f(y)$ .

For  $s_1 \in \{0, u, 1\}^n$  and  $s_2 \in \{0, u, 1\}^m$ , we denote as  $s_1 \cdot s_2$  the concatenation of  $s_1$  and  $s_2$ . For  $x \in \{0, u, 1\}^n$ , we define the *b-weight* of  $x$  as  $|x|_b := |\{i \in [n] : x_i = b\}|$  for some  $b \in \{0, u, 1\}$ .

Let  $g : \{0, u, 1\}^m \rightarrow \{0, u, 1\}$  be a natural function, and let  $X \in \{0, u, 1\}^{nm}$  be a ternary  $n \times m$  matrix. Then, we denote  $g[X] := (g(X_1), \dots, g(X_n))$  and  $dg[X; Y] := (dg(X_1; Y_1), \dots, dg(X_n; Y_n))$ , where  $X_i$  is the  $i$ 'th row of  $X$ .

A (De Morgan) Boolean circuit is a directed acyclic graph in which nodes of in-degree 0 are labeled with input variables, and each internal node is labeled with one of the Boolean operators  $\{\wedge, \vee, \neg\}$ , where the first two have fan-in 2 and the third has fan-in 1. Nodes of fan-out 0 are called outputs. An output gate computes a Boolean function in the natural way. A (De Morgan) formula is a circuit whose graph is a tree. As in Definition 1.5, we say that a formula is hazard-free if it has no hazards.

A circuit (formula) is monotone if it does not involve  $\neg$  gates. Each monotone circuit (formula) computes a monotone function, and each monotone function can be computed by a monotone circuit (formula). From now on, we speak only of De Morgan circuits and formulas, omitting the term ‘‘De Morgan’’.

The *size* of a Boolean circuit  $C$ , denoted  $\text{size}(C)$ , is defined as the number of vertices and edges in the underlying graph. The *depth* of a Boolean circuit  $C$ , denoted  $\text{depth}(C)$ , is defined as the length of the longest directed path from an input to an output gate. The size of a formula  $F$ , denoted  $\text{size}(F)$ , is defined as the number of leaves in the underlying tree.

We denote the minimal size of a circuit computing  $f$  by  $\text{size}_C(f)$ , and the minimal depth of a formula computing  $f$  by  $\text{depth}_F(f)$ . We add a superscript from  $\{u, +\}$  to the notation whenever we work in the hazard-free ( $u$ ) or the monotone ( $+$ ) setting, respectively. For example,  $\text{depth}_F^u(f)$  is the minimal depth of a hazard-free formula computing  $f$ .

Throughout the paper we use the hazard-free XOR operation, defined as follows:

**Definition 2.1** (XOR operation). *We use  $+$  to denote the XOR operator in the hazard-free setting. Let  $b, s \in \{0, u, 1\}$ , then:*

$$b + s := \begin{cases} 0, & b = s \text{ and } b, s \neq u, \\ 1, & b \neq s \text{ and } b, s \neq u, \\ u, & \text{otherwise.} \end{cases}$$

Computing  $x + y$  for  $x, y \in \{0, u, 1\}^n$  is done coordinatewise.

<sup>7</sup>Note the difference with Definition 1.2 in which 0 and 1 are incomparable, whereas here  $u$  is not included in the ordering.

## 2.1 Implicants and implicates

We prove the a useful property of  $P_1^{(f)}$  and  $P_0^{(f)}$  for a general Boolean function  $f$ . Later, we focus on monotone and unate functions.

**Proposition 2.2.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Let  $p_1 \in P_1^{(f)}$  and  $p_0 \in P_0^{(f)}$ . Let  $i, j \in [n]$  such that  $(p_1)_i \neq u$  and  $(p_0)_j \neq u$ . Then,*

- (1) *There exists  $q_0 \in P_0^{(f)}$  such that  $i$  is the only different and stable coordinate of  $q_0$  and  $p_1$ .*
- (2) *There exists  $q_1 \in P_1^{(f)}$  such that  $j$  is the only different and stable coordinate of  $q_1$  and  $p_0$ .*

PROOF. **TOPROVE 0**

□

**2.1.1 Monotone Boolean functions.** The implicants and implicates of a monotone Boolean function can be studied by considering only the maximal and minimal resolutions with respect to  $\leq$ :

**Definition 2.3** (The maximal and the minimal resolutions w.r.t  $\leq$ ). *Let  $x \in \{0, u, 1\}^n$ . We call  $\hat{x} := x|_{u \leftarrow 1}$  the maximal resolution of  $x$  with respect to  $\leq$  and  $\check{x} := x|_{u \leftarrow 0}$  the minimal resolution of  $x$  with respect to  $\leq$ .*

**Lemma 2.4** (Implicants and implicates of a monotone function). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Let  $p \in \{0, u, 1\}^n$ , then:*

- (1)  *$p \in I_1^{(f)}$  if and only if  $f(\check{p}) = 1$ .*
- (2)  *$p \in I_0^{(f)}$  if and only if  $f(\hat{p}) = 0$ .*

PROOF. **TOPROVE 1**

□

**Lemma 2.5** (Prime implicants and prime implicates of a monotone function). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Let  $p \in \{0, u, 1\}^n$  and denote  $S := \{i \in [n] : p_i \neq u\}$ , then:*

- (1)  *$p \in P_1^{(f)}$  if and only if  $f(\check{p}) = 1$  and for every  $i \in S$  it holds that  $f(\check{p}|_{i \leftarrow 0}) = 0$ .*
- (2)  *$p \in P_0^{(f)}$  if and only if  $f(\hat{p}) = 0$  and for every  $i \in S$  it holds that  $f(\hat{p}|_{i \leftarrow 1}) = 1$ .*

PROOF. **TOPROVE 2**

□

**Corollary 2.6.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function, then:*

- (1) *Every  $p_1 \in P_1^{(f)}$  satisfies  $p_1 \in \{1, u\}^n$ .*
- (2) *Similarly,  $P_0^{(f)} \subseteq \{0, u\}^n$ .*

PROOF. **TOPROVE 3**

□

**2.1.2 Unate Boolean functions.** The next lemmata generalize Corollary 2.6.

**Lemma 2.7** (The prime implicants of a unate Boolean function in  $x_i$ ). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then:*

- (1)  *$f$  is positive unate in variable  $x_i$  if and only if there is no  $p_1 \in P_1^{(f)}$  such that  $(p_1)_i = 0$ .*
- (2)  *$f$  is negative unate in variable  $x_i$  if and only if there is no  $p_1 \in P_1^{(f)}$  such that  $(p_1)_i = 1$ .*

PROOF. **TOPROVE 4**

□

**Lemma 2.8** (The prime implicates of a unate Boolean function in  $x_i$ ). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then:*

- (1)  *$f$  is positive unate in variable  $x_i$  if and only if there is no  $p_0 \in P_0^{(f)}$  such that  $(p_0)_i = 1$ .*
- (2)  *$f$  is negative unate in variable  $x_i$  if and only if there is no  $p_0 \in P_0^{(f)}$  such that  $(p_0)_i = 0$ .*

The proof of Lemma 2.8 is similar.

## 2.2 Communication complexity

In this section, we provide some basic notation and definitions from communication complexity. We assume familiarity with the basic concepts. See [22, 29] for textbooks on communication complexity.

Let  $X, Y, Z$  be finite sets. For a relation  $\mathcal{R} \subseteq X \times Y \times Z$ , we denote by  $CC(\mathcal{R})$  the minimal communication complexity required to solve  $\mathcal{R}$ . Any protocol  $P$  that solves a relation  $\mathcal{R} \subseteq X \times Y \times Z$  induces a partition of  $X \times Y$  into  $\mathcal{R}$ -monochromatic rectangles. The number of rectangles, denoted  $\text{mono-rec}(P)$ , is the number of the leaves in  $P$ . We denote by  $\text{mono-rec}(\mathcal{R})$  the minimum of  $\text{mono-rec}(P)$  among all protocols  $P$  that solve  $\mathcal{R}$ . We shall abuse notation and use  $CC(M)$  or  $\text{mono-rec}(M)$  instead of  $CC(\mathcal{R})$  or  $\text{mono-rec}(\mathcal{R})$  when  $M$  is a communication matrix of  $\mathcal{R}$ .

Let  $\mathcal{R} \subseteq X \times Y \times Z$  be a relation and let a protocol  $P$  for  $\mathcal{R}$ . Say there exists an input  $(x^*, y^*)$  for which there is only one valid output,  $z^* \in Z$ . Since  $P$  solves  $\mathcal{R}$ ,  $(x^*, y^*)$  reaches a leaf in the protocol tree that is labeled with  $z^*$ . The associated rectangle is referred as a  $z^*$ -uniform rectangle:

**Definition 2.9** ( $z^*$ -uniform combinatorial rectangle). *Let  $\mathcal{R} \subseteq X \times Y \times Z$  be a communication problem. Let  $\text{Rec}$  be a  $\mathcal{R}$ -monochromatic rectangle. We call  $\text{Rec}$  a  $z^*$ -uniform rectangle if there exists an entry  $(x^*, y^*) \in \text{Rec}$  such that  $\{z \in Z : (x^*, y^*, z) \in \mathcal{R}\} = \{z^*\}$ .*

Let  $\mathcal{R} \subseteq X \times Y \times Z$  and  $\mathcal{R}' \subseteq X' \times Y' \times Z'$  be relations. We say that  $\mathcal{R}$  is *reducible* to  $\mathcal{R}'$ , and denote  $\mathcal{R} \leq \mathcal{R}'$ , if there exist functions  $\phi_1 : X \rightarrow X'$ ,  $\phi_2 : Y \rightarrow Y'$  and  $\psi : Z' \rightarrow Z$  such that for every  $x \in X, y \in Y$  and  $z' \in Z'$ :

$$(\phi_1(x), \phi_2(y), z') \in \mathcal{R}' \Rightarrow (x, y, \psi(z')) \in \mathcal{R}.$$

It is clear that if  $\mathcal{R} \leq \mathcal{R}'$  then  $CC(\mathcal{R}) \leq CC(\mathcal{R}')$ .

We also need the notion of direct sum of two relations.

**Definition 2.10** (Direct sum of relations [17, Definition 2]). *Given two relations  $\mathcal{R} \subseteq X \times Y \times Z$  and  $\mathcal{R}' \subseteq X' \times Y' \times Z'$ , we define the direct sum of  $\mathcal{R}$  and  $\mathcal{R}'$  as:*

$$\mathcal{R} \otimes \mathcal{R}' \subseteq (X \times X') \times (Y \times Y') \times (Z \times Z'),$$

where  $((x_1, x_2), (y_1, y_2), (z_1, z_2)) \in \mathcal{R} \otimes \mathcal{R}'$  if and only if  $(x_1, y_1, z_1) \in \mathcal{R}$  and  $(x_2, y_2, z_2) \in \mathcal{R}'$ .

It is clear that if  $\mathcal{S} \leq \mathcal{R}$  and  $\mathcal{S}' \leq \mathcal{R}'$ , then  $\mathcal{S} \otimes \mathcal{S}' \leq \mathcal{R} \otimes \mathcal{R}'$  and  $CC(\mathcal{S} \otimes \mathcal{S}') \leq CC(\mathcal{R} \otimes \mathcal{R}')$ . The rank lower bound [22, Lemma 1.28] also implies that  $CC(\mathcal{R} \otimes \mathcal{R}) \geq \log(\text{Rank}(M_{\mathcal{R}})) + \log(\text{Rank}(M_{\mathcal{R}'}))$ , where  $M_{\mathcal{R}}$  and  $M_{\mathcal{R}'}$  are the communication matrices of  $\mathcal{R}$  and  $\mathcal{R}'$ , respectively.

**2.2.1 Simplification lemma for communication matrices.** We next present a lemma that will be important in our analysis of the communication matrices of the hazard-derivatives.

Let  $Z$  be a finite set. Denote by  $\mathbf{P}(Z)$  the power set of  $Z$ . Let  $M \in \mathbf{P}(Z)^{k \times \ell}$  and  $M' \in \mathbf{P}(Z)^{k' \times \ell}$  be matrices with entries in  $\mathbf{P}(Z)$  such that  $k \geq k'$ . Additionally, let  $X, Y$  be finite sets such that  $X, X' \subseteq X$  are the row labels and  $Y, Y' \subseteq Y$  are the column labels of  $M$  and  $M'$ , respectively. Note that the row and column labels of  $M$  and  $M'$  are allowed to be different. The matrices  $M$  and  $M'$  can be naturally associated with communication problems.

**Definition 2.11** (A sub-row and a super-row). *Let  $M$  and  $M'$  be matrices as in the above setting. Let  $i \in [k]$  and  $i' \in [k']$ . We say that a row  $M'_{i'}$  is a sub-row of  $M_i$  if the following holds:*

$$\forall j \in [\ell], \quad M'_{i',j} \subseteq M_{i,j}.$$

*We say that  $M_i$  is a super-row in  $M'$  if there exists some  $i' \in [k']$  such that  $M'_{i'}$  is a sub-row of  $M_i$ .*

**Lemma 2.12** (Communication matrix simplification). *Let  $M$  and  $M'$  be matrices as in the above setting. W.l.o.g, assume that the rows in  $M$  and the rows in  $M'$  are distinct. Moreover, assume:*

- (1) *For every  $i \in [k]$ ,  $M_i$  is a super-row in  $M'$ .*

(2) For every  $i' \in [k']$  there exists  $i \in [k]$  such that  $M'_{i'} = M_i$ .

Then,

$$\text{mono-rec}(M) = \text{mono-rec}(M'), \quad \text{CC}(M) = \text{CC}(M').$$

PROOF. **TOPROVE 5**

□

**2.2.2 Karchmer-Wigderson games.** Karchmer and Wigderson [18] showed an equivalence between the depth and size of a formula and the cost of an associated communication search problem, both in the general setting and the monotone setting.

**Definition 2.13** (KW game [18]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The Karchmer-Wigderson game of  $f$ , denoted  $KW_f$ , is the following communication problem: Alice receives  $x \in \{0, 1\}^n$  such that  $f(x) = 1$ , Bob receives  $y \in \{0, 1\}^n$  such that  $f(y) = 0$  and their goal is to determine a coordinate  $i \in [n]$  such that  $x_i \neq y_i$ .*

Note that for a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , the communication problem can be defined as the relation  $KW_f \subseteq X \times Y \times Z$  where  $X = f^{-1}(1)$ ,  $Y = f^{-1}(0)$  and  $Z = [n]$ .

**THEOREM 2.14** (COMMUNICATION VS. FORMULA [18]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then,*

$$\text{depth}_F(f) = \text{CC}(KW_f) \text{ and } \text{size}_F(f) = \text{mono-rec}(KW_f).$$

Similarly, a monotone version of the game is defined, with the distinction that Alice and Bob must output a coordinate where Alice's input has a 1 and Bob's input has a 0. This version of the game captures the complexity of monotone computation:

**Definition 2.15** (Monotone KW game [18]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. The Monotone Karchmer-Wigderson game of  $f$ , denoted  $KW_f^+$ , is the following communication problem: Alice receives  $x \in \{0, 1\}^n$  such that  $f(x) = 1$ , Bob receives  $y \in \{0, 1\}^n$  such that  $f(y) = 0$  and their goal is to determine a coordinate  $i \in [n]$  such that  $1 = x_i \neq y_i = 0$ .*

**THEOREM 2.16** (COMMUNICATION VS. MONOTONE FORMULA [18]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Then,*

$$\text{depth}_F^+(f) = \text{CC}(KW_f^+) \text{ and } \text{size}_F^+(f) = \text{mono-rec}(KW_f^+).$$

### 3 KW game of the hazard-derivative

In this section, we prove new insights on the  $KW^u$  communication matrices of the hazard-derivatives, while relying on the framework of hazard-free KW games restricted to the prime implicants and prime impicates (see Theorem 1.18). Our main observation is that for every  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $x \in \{0, 1\}^n$ , the communication matrix of  $KW_{df(x; \cdot)}^+$  is a submatrix of  $M_{KW_f^u}$ , that captures the communication complexity of a (potentially larger) submatrix of  $M_{KW_f^u}$ .

We present a simple example which serves as an illustration throughout this section. Consider  $f_{\text{prime}} : \{0, 1\}^4 \rightarrow \{0, 1\}$ , defined such that for every  $k \in \mathbb{N}$ ,  $0 \leq k \leq 15$ ,  $f_{\text{prime}}(\text{bin}(k)) = 1$  if and only if  $k$  is a prime number. One can calculate the prime implicants and prime impicates of  $f_{\text{prime}}$  and sketch its hazard-free KW game communication matrix, as demonstrated in Figure 2a.

Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function and let  $x \in \{0, 1\}^n$  be a fixed Boolean input such that  $f(x) = b \in \{0, 1\}$ . We begin by proving the following connection between the prime implicants and prime impicates of  $f$  and  $df(x; \cdot)$ :

**Lemma 3.1** (The prime impicates of the hazard-derivative).

$$P_0^{(df(x; \cdot))} = \{p_b + x : p_b \in P_b^{(f)}|_x\}.$$

	uu00	u1u0	1uu0	u00u	111u
001u	3	2	1	3	1, 2
u101	4	4	4	2	3
u011	3, 4	2, 4	4	3	2
0u11	3, 4	4	1, 4	3	1
01u1	4	4	1, 4	2	1

(a) The matrix of  $f_{\text{prime}}$ .

uu00	u0u0	0uu0	u00u	000u
111u	3	2	1	
uuu1	4	4	4	

u00u	u1uu	uu1u
2		
3		

000u	uu1u	u1uu	1uuu
3			
2			
1			

(b) The matrix of  $\text{df}_{\text{prime}}(1100; \cdot)$ .(c) The matrix of  $\text{df}_{\text{prime}}(1001; \cdot)$ .(d) The matrix of  $\text{df}_{\text{prime}}(1111; \cdot)$ .Fig. 2. Sketch of  $\text{KW}^u$  communication matrices for  $f_{\text{prime}}$ .

**Lemma 3.2** (The prime implicants of the hazard-derivative).

$$P_1^{(\text{df}(x; \cdot))} \subseteq \{(p_{-b} + x)|_{0 \Leftarrow u} : p_{-b} \in P_{-b}^{(f)}\} \subseteq I_1^{(\text{df}(x; \cdot))}.$$

To demonstrate the claims consider Figure 2b. For the Boolean input 1100, it holds that  $f_{\text{prime}}(1100) = 0$ . It can be verified that  $P_0^{(f_{\text{prime}})}|_{1100} = \{uu00, u1u0, 1uu0\}$ . By Lemma 3.1, the prime implicants of  $\text{df}_{\text{prime}}(1100; \cdot)$  are  $\{uu00 + 1100, u1u0 + 1100, 1uu0 + 1100\} = \{uu00, u0u0, 0uu0\}$ . This result makes sense: apart from 0010, even numbers are not prime. To transform 1100 to 0010, all three of its most significant bits must be perturbed. Therefore, any perturbation affecting at most two of these three bits preserves the function's value. Perturbing the least significant bit yields 1101, the prime number 13, so it must remain unchanged. Following Lemma 3.2, calculating  $(p_1 + 1100)|_{0 \Leftarrow u}$  for every  $p_1 \in P_1^{(f_{\text{prime}})}$  yields  $\{111u, u111, 1u11, 1uu1, uuu1\}$ . This set does not consist solely of *prime* implicants, as  $u111, 1u11, 1uu1$  are already “covered” by  $uuu1$ . Furthermore, this is a subset of  $I_1^{(\text{df}_{\text{prime}}(1100; \cdot))}$  as all but 111u perturb the least significant bit of 1100, and  $R(110u)$  contains 1101. Similarly, 111u perturbs (at least) the three most significant bits, and  $R(uuu0)$  contains 0010, which is the prime number 2.

Let us discuss the underlying intuition before providing the full proofs. The hazard-derivative is a monotone function and so it is enough to consider the maximal and the minimal resolutions with respect to  $\leq$  (see Section 2.1.1 for details).

Consider  $q_0 \in P_0^{(\text{df}(x; \cdot))}$ . The 1s in  $\hat{q}_0$  represent all the coordinates in  $x$  that *can* be perturbed while the output  $\tilde{f}(x + u\hat{q}_0)$  still equals  $f(x) = b$ . Since  $q_0$  is prime, it should not be possible to add more us to it, and consequently to  $x + u\hat{q}_0$ . This is the same as saying that  $x + u\hat{q}_0 = p_b \in P_b^{(f)}|_x$ . Since  $u\hat{q}_0 = q_0$ , it follows that  $q_0 = x + p_b$ . The other direction is more intuitive: for  $p_b \in P_b^{(f)}|_x$ , the 0's in  $x + p_b$  encode exactly the stable coordinates in  $p_b$ . This is a minimal subset of coordinates that must *not* be perturbed in  $x$  if we wish to get the output  $f(x)$ .

Consider  $q_1 \in P_1^{(\text{df}(x; \cdot))}$ . The 1s in  $\hat{q}_1$  represent all the coordinates in  $x$  that *must* be perturbed in order for  $\tilde{f}(x + u\hat{q}_1)$  to be equal  $u$ . Hence, there must be a resolution  $z \in R(x + u\hat{q}_1)$  for which  $f(z) \neq f(x)$ . We prove in Lemma 3.3 that for every  $q_1 \in P_1^{(\text{df}(x; \cdot))}$  such  $z$  is unique among  $R(x + u\hat{q}_1)$ . Specifically,  $z$  is the resolution that differs from  $x$  in the the maximal number of coordinates possible,

that is, in all coordinates for which  $\check{q}_1$  equals 1. Since  $f(z) = \neg b$  there is some  $p_{\neg b} \in P_{\neg b}^{(f)}$  such that  $p_{\neg b} \stackrel{u}{\leq} z$ . We then show that  $q_1 = (p_{\neg b} + x)|_{0 \Leftarrow u}$ , relying on the fact that  $z$  is unique.

PROOF. **TOPROVE 6** □

Before giving a full proof for Lemma 3.2, we prove the following auxiliary lemma:

**Lemma 3.3.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function, and let  $x \in \{0, 1\}^n$  be a fixed Boolean input such that  $f(x) = b \in \{0, 1\}$ . Then,  $q_1 \in P_1^{(df(x; \cdot))}$  if and only if  $q_1 \in \{1, u\}^n$  and the following  $z \in R(x + u\check{q}_1)$  is the only resolution for which  $f(z) = \neg b$ :*

$$z := \forall i \in [n], \quad (z)_i = \begin{cases} x_i, & (\check{q}_1)_i = 0, \\ \neg x_i, & (\check{q}_1)_i = 1. \end{cases}$$

PROOF. **TOPROVE 7** □

PROOF. **TOPROVE 8** □

**Remark 3.4.** *Note that unlike Lemma 3.1, in Lemma 3.2 we do not get a complete characterization of the prime implicants of the hazard-derivative. There may be  $p_{\neg b} \in P_{\neg b}^{(f)}$  such that for  $p := (p_{\neg b} + x)|_{0 \Leftarrow u}$  there are multiple  $z \in R(x + u\check{p})$  such that  $f(z) \neq f(x)$ . In this case, we prove that  $p$  is an implicant of  $df(x; \cdot)$  (but not prime). See Example 3.5.*

**Example 3.5** (In Lemma 3.2 equality does not necessarily hold). *We present a simple example where  $P_1^{(df(x; \cdot))} \neq \{(p_{\neg b} + x)|_{0 \Leftarrow u} : p_{\neg b} \in P_{\neg b}^{(f)}\}$ . Consider  $f : \{0, 1\}^3 \rightarrow \{0, 1\}$  such that  $f^{-1}(1) = \{000, 001, 011, 111\}$ . It is not hard to see that  $P_1^{(f)} = \{u11, 0u1, 00u\}$ . When setting  $x := 110$ , we get  $b := f(x) = 0$  and:*

$$\{(p_{\neg b} + x)|_{0 \Leftarrow u} : p_{\neg b} \in P_{\neg b}^{(f)}\} = \{uu1, 1u1, 11u\}.$$

*Clearly,  $1u1$  is not a prime implicant of  $df(x; \cdot)$ , since  $uu1$  is also an implicant. Moreover, both  $001, 011 \in R(uu1)$ . Therefore,  $\{uu1, 1u1, 11u\} \neq P_1^{(df(x; \cdot))}$ .*

We showed how to infer  $P_1^{(df(x; \cdot))}$  and  $P_0^{(df(x; \cdot))}$  from  $P_1^{(f)}$  and  $P_0^{(f)}$ . In fact, every  $q_1 \in P_1^{(df(x; \cdot))}$  is produced by one (or more)  $p_{\neg b} \in P_{\neg b}^{(f)}$ , and every  $q_0 \in P_0^{(df(x; \cdot))}$  is produced by exactly one  $p_b \in P_b^{(f)}$ . This translation preserves the set of stable and different coordinates. As a result, these sets are the equal for  $p_{\neg b}, p_b$  and  $q_1, q_0$ .

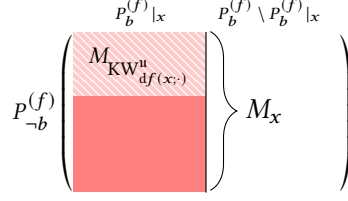
**Lemma 3.6.** *Let  $x \in \{0, 1\}^n$  such that  $f(x) = b \in \{0, 1\}$ . Let  $q_1 \in P_1^{(df(x; \cdot))}$  and  $q_0 \in P_0^{(df(x; \cdot))}$ . According to Lemma 3.2 and Lemma 3.1 there exist  $p_b \in P_b^{(f)}|_x$  and  $p_{\neg b} \in P_{\neg b}^{(f)}$  such that  $q_1 = (p_{\neg b} + x)|_{0 \Leftarrow u}$  and  $q_0 = p_b + x$ . Then, for every  $i \in [n]$  the following holds:*

$$(q_1 + q_0)_i = 1 \iff (p_{\neg b} + p_b)_i = 1.$$

PROOF. **TOPROVE 9** □

A concrete example of Lemma 3.6 is shown in Figures 2a and 2b, where the same colors as in Figure 3 are used for  $M_{1100}$  and  $M_{KW_{df_{\text{prime}}(1100; \cdot)}}^u$ . It can be verified that  $u1u0 \in P_0^{(f_{\text{prime}})}$  yields  $u0u0 \in P_0^{(df_{\text{prime}}(1100; \cdot))}$  and  $001u \in P_1^{(f_{\text{prime}})}$  yields  $111u \in P_1^{(df_{\text{prime}}(1100; \cdot))}$  with the corresponding entries both equal to  $\{2\}$ .



Fig. 3.  $M_{KW_f^u}$ ,  $M_x$  and  $M_{KW_{df(x;\cdot)}^u}$ .

Let  $M_{KW_f^u}$  and  $M_{KW_{df(x;\cdot)}^u}$  denote the communication matrix of the hazard-free KW game of  $f$  and  $df(x; \cdot)$ , respectively. Let  $M_x$  be the submatrix of  $M_{KW_f^u}$  with labels  $p_{\neg b}^{(f)} \times p_b^{(f)}|_x$ . We prove the following, as illustrated in Figure 3:

**Proposition 3.7.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function. For every  $x \in \{0, 1\}^n$ ,  $M_{KW_{df(x;\cdot)}^u}$  is a submatrix of  $M_{KW_f^u}$ .*

PROOF. TOPROVE 10 □

**Remark 3.8.** *Observe that in the proof of Proposition 3.7, for every  $j \in [\ell]$  it holds that  $p_b^{(\sigma(j))} \in p_b^{(f)}|_x$ . Thus,  $M_{KW_{df(x;\cdot)}^u}$  is actually a submatrix of  $M_x$ .*

Proposition 3.7 can be viewed as an alternative proof for Theorem 1.12 for formulas:

**Corollary 3.9.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function and let  $x \in \{0, 1\}^n$  be a fixed Boolean input. It holds that:*

$$\text{size}_F^+(\text{df}(x; \cdot)) \leq \text{size}_F^u(f).$$

PROOF. TOPROVE 11 □

Next, we make the simple but important observation. Certain rows, which we refer to as “super-rows”, can be removed from a communication matrix without effecting its complexity. A *super-row* is a row  $r$  in the matrix such that there exists another row  $r'$ , called a *sub-row*, whose every entry is a subset of the matching entry in  $r$  (see Section 2.2.1 for details).

It is not too difficult to see that the rows in  $M_x$  that are associated with  $p_{\neg b} \in p_{\neg b}^{(f)}$  such that  $(p_{\neg b} + x)|_{0 \leftarrow u} \in I_1^{(\text{df}(x; \cdot))} \setminus P_1^{(\text{df}(x; \cdot))}$  are super-rows in  $M_x$ . Implicants that are not prime, by definition, contain “extra” stable bits that can be replaced with  $u$  while preserving the output of the function. Therefore, these rows contain possibly more valid answers to the  $KW^u$  on  $df(x; \cdot)$  than the ones associated with  $P_1^{(\text{df}(x; \cdot))}$ . By Lemma 3.6, we conclude that these rows are equal to those of  $M_x$ , and thus we are done. Figures 2a and 2b demonstrate this claim, as the super-rows in 2a correspond to  $\{u011, 0u11, 01u1\}$ , which yielded  $\{u111, 1u11, 1uu1\}$  in our previous calculation.

Following our observation, the super-rows can be removed without changing the complexity of  $M_x$ . Most importantly, removing them results in  $M_{KW_{df(x;\cdot)}^u}$ . Hence,  $M_x$  and  $M_{KW_{df(x;\cdot)}^u}$  are equivalent in terms of communication complexity. This highlights the importance of the hazard-derivative, which captures the communication complexity of  $M_x$ , which is a potentially larger submatrix of  $M_{KW^u}$ .

**Proposition 3.10.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $x \in \{0, 1\}^n$  such that  $f(x) = b \in \{0, 1\}$ . We denote by  $M_x$  the submatrix of  $M_{KW_f^u}$  with labels  $P_{\neg b}^{(f)} \times P_b^{(f)}|_x$ . The following holds:*

$$\text{mono-rec}(M_x) = \text{mono-rec}(M_{KW_{df(x;\cdot)}^u}), \quad CC(M_x) = CC(M_{KW_{df(x;\cdot)}^u}).$$

PROOF. **TOPROVE 12**

□

#### 4 Tightness of the hazard-derivative method

In this section, we extend the hazard-derivative lower bound method of [12] when applied to hazard-free formulas. We show that the hazard-derivative method yields *exact* bounds if and only if the function is unate, thus proving Theorem 1.31. Hence, whether a function is *unate* or not, can serve as a criterion to achieving a monotone gap of 1, which in particular answers Question 1.13.

We prove Proposition 4.1 by applying Proposition 3.10, which immediately shows that the hazard-derivative lower bound method is tight for unate functions.

**Proposition 4.1** (Unate functions have monotone gap of 1). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant unate Boolean function. Then, there exists  $x \in \{0, 1\}^n$  such that:*

$$\text{size}_F^+(\text{df}(x; \cdot)) = \text{size}_F^u(f).$$

PROOF. **TOPROVE 13**

□

**Remark 4.2.** *An alternative proof of Proposition 4.1 is to generalize [12, Corollary 4.5]. A function  $f$  is unate if and only if there exists some  $c \in \{0, 1\}^n$  such that  $f(x + c)$  is monotone. Therefore, it can be shown by [12, Lemma 4.4], that  $f(x + c) = \text{df}(c; x)$ .*

The proof of the other direction requires more work. Consider a non-unate function  $f$  and a coordinate  $d \in [n]$  that demonstrates that  $f$  is not unate. As before, let  $x \in \{0, 1\}^n$  be a Boolean input such that  $f(x) = b \in \{0, 1\}$ . Let  $s \in \{0, u, 1\}$ , and denote by  $P_1^{d=s}$  ( $P_0^{d=s}$ ) the subset of  $P_1^{(f)}$  ( $P_0^{(f)}$ ) for which the  $d$ 'th coordinate equals  $s$ . By examining the prime implicants and the prime implicates of  $f$  we observe that  $P_b^{d=x_d}, P_b^{d=\neg x_d}, P_{\neg b}^{d=x_d}, P_{\neg b}^{d=\neg x_d} \neq \emptyset$ . We actually prove the following stronger claim, as illustrated in Figure 4:

**Proposition 4.3.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-unate Boolean function. Then, there exists a coordinate  $d \in [n]$ ,  $p_1, q_1 \in P_1^{(f)}$  and  $p_0, q_0 \in P_0^{(f)}$  such that:*

- (1)  $(p_1)_d = 0, (p_0)_d = 1$  and  $d$  is the only different and stable coordinate between  $p_1$  and  $p_0$ .
- (2)  $(q_1)_d = 1, (q_0)_d = 0$  and  $d$  is the only different and stable coordinate between  $q_1$  and  $q_0$ .

PROOF. **TOPROVE 14**

□

By Proposition 4.3, we conclude that for every protocol solving  $KW_f^u$ , there exists a monochromatic rectangle  $\text{Rec} \subseteq P_{\neg b}^{d=x_d} \times P_b^{d=\neg x_d}$  in the partition induced by the protocol that must be colored with  $d$  and is disjoint from  $M_x$ , as illustrated in Figure 5. Removing the leaf associated with  $\text{Rec}$  from the protocol tree results in a protocol for  $M_x$  that has one less leaf. This implies that there must be some gap in the number of leaves in the optimal protocols, thus proving the following proposition:

**Proposition 4.4.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-unate Boolean function. Then for every  $x \in \{0, 1\}^n$ , it holds that:*

$$\text{size}_F^+(\text{df}(x; \cdot)) < \text{size}_F^u(f).$$

PROOF. **TOPROVE 15**

□

The proof of Theorem 1.31 follows immediately from Proposition 4.1 and Proposition 4.4.

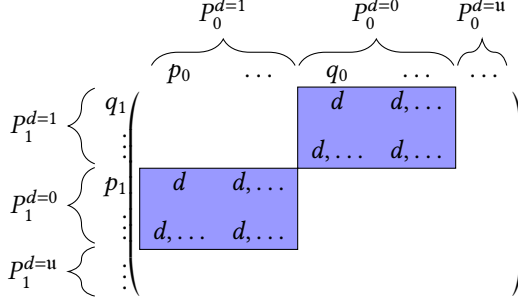


Fig. 4. Sketch of  $M_{KW_f^u}$  for a non-unate  $f$ . Empty entries do not contain the  $d$ 'th coordinate. We use the notation ' $d, \dots$ ' when there may be stable coordinates other than  $d$  on which the row labels and column labels differ.

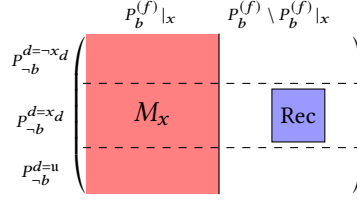


Fig. 5.  $M_{KW_f^u}$  for a non-unate  $f$ .

#### 4.1 The monotone gap of Andreev's function

In this section, we prove that Andreev's function gives monotone gap of  $\Omega\left(\frac{n^2}{\log n}\right)$ .

**Definition 4.5** (Andreev's function [2]). *Andreev's function, with parameters  $k, m \in \mathbb{N}$ , is a Boolean function over  $n = 2^k + km$  variables:*

$$\text{ANDREEV}_{k,m} : \{0, 1\}^{2^k} \times \{0, 1\}^{km} \rightarrow \{0, 1\}.$$

*Andreev's function receives as inputs  $f \in \{0, 1\}^{2^k}$  and  $X \in \{0, 1\}^{km}$ , a truth table of a Boolean function over  $k$  bits and a Boolean  $k \times m$  matrix, respectively:*

$$\text{ANDREEV}_{k,m}(f, X) := (f \diamond \text{XOR}_m)(X),$$

*where  $\text{XOR}_m : \{0, 1\}^m \rightarrow \{0, 1\}$  is the parity function over  $m$  bits:*

$$\text{XOR}_m(x) := x_1 + \dots + x_m.$$

In the rest of this section, we shall use the following parameters  $n := 2^k + km$ ,  $k := \log(n/2)$ , and  $m = \frac{n}{2 \log(n/2)}$ .<sup>8</sup>

Håstad was the first to prove a near-cubic lower bound on the formula complexity of Andreev's function [9] using the random restriction technique. Later, Tal [35] improved the analysis to obtain the currently best lower bound of  $\Omega\left(\frac{n^3}{\log^2 n \log \log n}\right)$ . This lower bound is tight up to the  $\log \log n$  factor.

<sup>8</sup>We can take  $k$  and  $n$  to be powers of 2 so we ignore ceiling and floors for readability.

THEOREM 4.6 (ANDREEV'S FUNCTION REQUIRES CUBIC FORMULAS [35, THEOREM 7.2]).

$$\Omega\left(\frac{n^3}{\log^2 n \log \log n}\right) \leq \text{size}_F(\text{ANDREEV}_{k,m}) \leq O\left(\frac{n^3}{\log^2 n}\right).$$

As  $\text{size}_F^u(f) \geq \text{size}_F(f)$ , this lower bound also holds in the hazard-free setting. We next bound the complexity of the hazard-derivatives of Andreev's function.

**Fact 4.7** (Parity has linear hazard-derivatives). *For every  $x, y \in \{0, 1\}^m$ , it holds that*

$$\text{dXOR}_m(x; y) = \bigvee_{i=1}^m y_i.$$

We denote  $\text{dXOR}_m(y) := \text{dXOR}_m(x; y)$ , since the hazard-derivative of  $\text{XOR}_m$  w.r.t to  $x$  does not depend on  $x$ .

**Claim 4.8** (Andreev's function has near linear derivatives). *For every  $f \in \{0, 1\}^{2^k}$  and  $X \in \{0, 1\}^{km}$ , it holds that:*

$$\text{size}_F^+(\text{dANDREEV}_{k,m}(f, X; \cdot, \cdot)) = (k+1)2^k + km = O(n \log(n)).$$

PROOF. **TOPROVE 16** □

**Corollary 4.9** (Andreev's function has near quadratic monotone gap).

$$\text{mono-gap}(\text{ANDREEV}_{k,m}) = \Omega\left(\frac{n^2}{\log n}\right).$$

PROOF. **TOPROVE 17** □

## 5 A universal formula upper bound for most Boolean functions

In this section, we study the hazard-free formula complexity of random Boolean functions.

**Definition 5.1.** *We say that  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is a random function if it is sampled uniformly at random among all Boolean functions on  $n$  variables. Alternatively, we can think as generating  $f$  by sampling, for every  $x \in \{0, 1\}^n$ , a value  $f(x) \in \{0, 1\}$  uniformly and independently.*

It is well known that the formula complexity of a random function is maximal.

THEOREM 5.2 ([33, 34]). *For any  $\epsilon > 0$ , almost all Boolean functions on  $n$  variables require formula size of at least  $\frac{(1-\epsilon)2^n}{\log n}$ .*

Since  $\text{size}_F^u(f) \geq \text{size}_F(f)$ , an immediate consequence of Theorem 5.2 is that with high probability, a random Boolean function over  $n$  variables requires hazard-free formulas of size at least  $\Omega(\frac{2^n}{\log n})$ .

In this section, we provide two proofs for Theorem 1.34, which greatly improves upon the best-known upper bound for arbitrary functions, currently  $O(3^n)$  (see Sections 5.1 and 5.3). The two proofs utilize the same observation: the probability that a random function is constant on any large Boolean subcube is small. Therefore, it tends to have prime implicants that consist mostly of stable bits, while its hazard-derivatives tend to be nearly constant and equal to 1. The second proof consists of proving Theorem 1.36, which gives a weak converse to Theorem 1.12 and offers insights of independent interest.

Another consequence of this observation is that the hazard-derivatives of random Boolean functions are significantly simpler than the functions themselves. This reveals a considerable gap between the hazard-free complexity of a random function and the monotone complexity of its hazard-derivatives. Consequently, random functions achieve a monotone gap that is exponential in  $n$ .

### 5.1 Hazard-free DNF formula of a random Boolean function

We show that, with high probability, all the prime implicants of a random function consist mostly of stable bits. Hence, their number is  $2^{(1+o(1))n}$  and a “brute force” DNF formula construction yields Theorem 1.34.

**Lemma 5.3** ( $P_1^{(f)}$  of a random  $f$  tend to have more stable bits). *Let  $p_1 \in \{0, u, 1\}^n$  be fixed such that  $|p_1|_u = k$ . The probability that  $p_1 \in P_1^{(f)}$ , for a random Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , is at most  $(\frac{1}{2})^{2^k}$ .*

PROOF. **TOPROVE 18**

□

**Lemma 5.4** (Hazard-free DNF formula for a random function). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a random Boolean function. Then, with high probability,  $f$  can be calculated by a hazard-free DNF formula of size at most  $\frac{n^{\log n}}{\log n^{(1-o(1)) \log n}} \cdot 2^n$ .*

PROOF. **TOPROVE 19**

□

### 5.2 The hazard-derivative of a random Boolean function

An argument similar to the one in the previous section, shows that the hazard-derivatives of a randomly sampled function have low monotone complexity. Hazard-derivatives indicate whether the function’s output changes when perturbing a fixed input. As more bits in the input are perturbed, the probability that the function’s output would not change gets smaller, since there are a lot of resolutions. Consequently, the derivative’s output is likely to be 1, making the hazard-derivative nearly constant.

**Lemma 5.5** ( $P_1^{(\text{df}(x; \cdot))}$  for a random  $f$  tend to have more unstable bits). *Let  $x \in \{0, 1\}^n$  and let  $q_1 \in \{0, u, 1\}^n$  be fixed such that  $|q_1|_1 = k$ . The probability that  $q_1 \in P_1^{(\text{df}(x; \cdot))}$ , for a random Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , is  $(\frac{1}{2})^{2^k-1}$ .*

PROOF. **TOPROVE 20**

□

**Lemma 5.6.** *A random Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , sampled uniformly at random, satisfies that with high probability, for every  $x \in \{0, 1\}^n$ :*

$$\text{size}_F^+(\text{df}(x; \cdot)) \leq \frac{n^{\log n}}{\log n^{(1-o(1)) \log n}}.$$

PROOF. **TOPROVE 21**

□

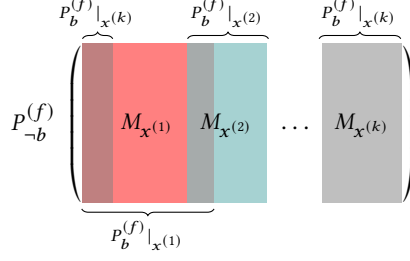
The proof of Proposition 1.35, which we restate for ease of reading, follows immediately.

**Proposition 1.35.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a random Boolean function. Then, with high probability,*

$$\text{mono-gap}(f) = \omega\left(\frac{2^n}{n^{\log n}}\right).$$

PROOF. **TOPROVE 22**

□

Fig. 6. Decomposition of  $M_{KW_f^u}$ .

### 5.3 Hazard-derivatives upper bound

In this section, we prove Theorem 1.36, which upper bounds the hazard-free formula size in terms of the monotone formula size of several of its hazard-derivatives. Building on the analysis of  $M_{KW_f^u}$  discussed in Section 3, we prove that  $M_{KW_f^u}$  can be decomposed into submatrices, each with communication complexity bounded from above by that of a specific hazard-derivative. When given a subset  $\{x^{(1)}, \dots, x^{(k)}\}$  of boolean inputs in  $X_b \subseteq \{0, 1\}^n$  for some  $b \in \{0, 1\}$  that are enough to cover  $P_b$  (i.e., for every  $p \in P_b$  there exists  $x \in X_b$  such that  $p \preceq^u x$ ), we decompose  $M_{KW_f^u}$  to  $k$  (potentially overlapping)  $M_x$ 's with labels  $P_{-b}^{(f)} \times P_b^{(f)}|_x$ , as illustrated in Figure 6 (see Figure 2 for a concrete example).

We leverage Theorem 1.36 to prove Theorem 5.7. Thus, we provide an alternative proof for the improved upper bound on the hazard-free formula complexity of random Boolean functions (see Theorem 1.34). Furthermore, Theorem 5.7 demonstrates that the upper bound in Theorem 1.36 is not too far from the truth for random Boolean functions. In Section 5.3.1 we provide an example of an explicit function for which Theorem 1.36 yields a tight bound.

To ease the reading, we repeat the statement of Theorem 1.36.

**THEOREM 1.36 (HAZARD-DERIVATIVES UPPER BOUND).** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $b \in \{0, 1\}$ . Let  $X_b \subseteq f^{-1}(b)$ . Then,*

$$\text{if } P_b^{(f)} = \bigcup_{x \in X_b} P_b^{(f)}|_x, \text{ then it holds that } \text{size}_F^u(f) \leq \sum_{x \in X_b} \text{size}_F^+(df(x; \cdot)).$$

**PROOF.** TOPROVE 23

□

As a corollary, we get a universal upper bound on the hazard-free formula complexity of random Boolean functions, which in particular proves Theorem 1.34.

**THEOREM 5.7 (A UNIVERSAL UPPER BOUND FOR MOST BOOLEAN FUNCTIONS).** *For a random Boolean function  $f$ , we get that with high probability, for a minimally sized  $X$  such that*

$$P_0^{(f)} = \bigcup_{x \in X} P_0^{(f)}|_x,$$

*the following holds:*

$$\left( \sum_{x \in X} \text{size}_F^+(df(x; \cdot)) \right)^{1-o(1)} \leq \text{size}_F^u(f) \leq \sum_{x \in X} \text{size}_F^+(df(x; \cdot)).$$

*In particular, for a random Boolean function  $f$ , with high probability, we get that:*

$$\frac{2^n}{\log n} \leq \text{size}_F^u(f) \leq 2^n \cdot \frac{n^{\log n}}{\log n^{(1-o(1)) \log n}}.$$

PROOF. **TOPROVE 24**

□

**5.3.1 Tight bound for range functions.** Here, we prove that Theorem 1.36 can give exact bounds. We consider the special case where there exists  $x^* \in \{0, 1\}^n$  such that  $P_0^{(f)} = P_0^{(f)}|_{x^*} \cup P_0^{(f)}|_{\neg x^*}$ . The key observation is that in this case, the prime implicants are covered by  $x^*$  and  $\neg x^*$ . Since  $x^*$  and  $\neg x^*$  are negated, all stable coordinates of  $P_0^{(f)}|_{x^*}$  and  $P_0^{(f)}|_{\neg x^*}$  are different, making the row entries of  $M_{x^*}$  and  $M_{\neg x^*}$  disjoint. Therefore, there is an optimal protocol that begins by dividing the communication matrix to  $M_{x^*}$  and  $M_{\neg x^*}$ . Later, we discuss a family of explicit functions that satisfy the above property.

**Proposition 5.8.** *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function. If there exists  $x^* \in \{0, 1\}^n$  such that  $P_0^{(f)} = P_0^{(f)}|_{x^*} \cup P_0^{(f)}|_{\neg x^*}$ , then:*

$$\text{size}_F^u(f) = \text{size}_F^+(df(x^*; \cdot)) + \text{size}_F^+(df(\neg x^*; \cdot)).$$

PROOF. **TOPROVE 25**

□

We next present an example of an explicit function that satisfies the assumption of Proposition 5.8.

**Definition 5.9** (Threshold and Range functions). *Let  $k \in [n]$ . A threshold- $k$  function  $T_k^n : \{0, 1\}^n \rightarrow \{0, 1\}$  is defined such that  $T_k^n = 1$  iff  $x_1 + \dots + x_n \geq k$ .*

*Let  $a, b \in [n]$  such that  $0 < a < b \leq n$ . We call  $R_{a,b}^n$  an  $a$ - $b$ -range function, if for every  $x \in \{0, 1\}^n$ ,  $R_{a,b}^n(x) = 1$  iff  $a \leq x_1 + \dots + x_n < b$ . In other words,  $R_{a,b}^n(x) = T_a^n(x) \wedge \neg T_b^n(x)$ .*

**Lemma 5.10** (Prime implicants and prime impicates of range functions). *Let  $R_{a,b}^n : \{0, 1\}^n \rightarrow \{0, 1\}$  be an  $a$ - $b$ -range function. Let  $p \in \{0, 1\}^n$ , then:*

- (1)  $p \in P_1^{(R_{a,b}^n)}$  if and only if  $|p|_1 = a$  and  $|p|_0 = n - (b - 1)$ .
- (2)  $p \in P_0^{(R_{a,b}^n)}$  if and only if  $|p|_0 = 0$  and  $|p|_1 = b$  or  $|p|_1 = 0$  and  $|p|_0 = n - (a - 1)$ .

PROOF. **TOPROVE 26**

□

**Lemma 5.11.** *Let  $R_{a,b}^n : \{0, 1\}^n \rightarrow \{0, 1\}$  be an  $a$ - $b$ -range function, then:*

$$\begin{aligned} dR_{a,b}^n(\bar{0}; \cdot) &= T_a^n, \\ dR_{a,b}^n(\bar{1}; \cdot) &= T_{n-(b-1)}^n. \end{aligned}$$

PROOF. **TOPROVE 27**

□

**Proposition 5.12** (Tight lower bound for range functions). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be an  $a$ - $b$ -range function, then:*

$$\text{size}_F^u(f) = \text{size}_F^+(T_a^n) + \text{size}_F^+(T_{n-(b-1)}^n).$$

PROOF. **TOPROVE 28**

□

## 6 Composition of Boolean functions in hazard-free setting

In this section, we prove Proposition 1.37, which establishes a lower bound on the communication complexity of the hazard-free KW game for the composition of a monotone function with an arbitrary inner function. We follow the proof idea of the next lemma:

**Lemma 6.1** ([17, Lemma 4]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be monotone Boolean functions. Then:*

$$KW_f^+ \otimes KW_g^+ \leq KW_{f \circ g}^+.$$

The proof of Lemma 6.1 uses two properties of  $KW_{f \diamond g}^+$ : First, the game can be played equivalently on the prime implicants and the prime implicates of the function, and those of  $f \diamond g$  are constructed of those of  $f$  and those of  $g$ . Second, for an input  $X, Y \in \{0, u, 1\}^{n \times m}$ , a valid answer  $(i, j)$  to  $KW_{f \diamond g}^+$  must be a row  $i$  for which  $g(X_i) = 1, g(Y_i) = 0$  and so  $i$  is a valid answer to the KW game of  $f$  and  $j$  is a valid answer to the KW game of  $g$ . The lower bound is then derived through a reduction to the direct sum of functions, as shown in [17, Corollary 5]. We prove that these properties also hold in the hazard-free KW game of  $f \diamond g$  if  $f$  is a monotone function, while  $g$  can be any function.

We prove in Proposition 6.2 that the prime implicants (and implicates) of the composed function,  $f \diamond g$ , can be derived from the prime implicants (and implicates) of  $f$  and  $g$ .

**Proposition 6.2** (Prime implicants of composition). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be Boolean functions. Then  $P \in P_1^{(f \diamond g)} \subseteq \{0, u, 1\}^{nm}$  if and only if:*

- (1)  $\tilde{g}[P] \in P_1^{(f)}$ .
- (2) For every  $i \in [n]$ , if  $\tilde{g}(P_i) = b \in \{0, 1\}$  then  $P_i \in P_b^{(g)}$ . Otherwise,  $P_i = u^m$ .

In order to prove Proposition 6.2 we would need the following lemma, which shows that the hazard-free extension of the block composition of two Boolean functions is the same as the block composition of their hazard-free extensions. We define the block composition of ternary functions exactly as in the Boolean case.

**Lemma 6.3** (Hazard-free extension of composition). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be Boolean functions. Then:*

$$\widetilde{f \diamond g} = \tilde{f} \diamond \tilde{g}.$$

We begin by proving Proposition 6.2 and later prove Lemma 6.3.

PROOF. TOPROVE 29 □

For the proof of Lemma 6.3 we require the next simple lemma.

**Lemma 6.4.** *Let  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be a ternary function. Let  $P \in \{0, u, 1\}^{nm}$  be a ternary  $n \times m$  matrix. Then:*

$$R(\tilde{g}[P]) = \{g[Z] : Z \in R(P)\}.$$

PROOF. TOPROVE 30 □

PROOF. TOPROVE 31 □

Given Proposition 6.2, if  $P_1^{(f)} \subseteq \{1, u\}^n$  and  $P_0^{(f)} \subseteq \{0, u\}^n$ , a valid answer to the hazard-free KW game of  $f \diamond g$  on the inputs  $X, Y \in \{0, u, 1\}^{nm}$  must lie in a row  $i$  such that  $X_i \in P_1^{(g)}$  and  $Y_i \in P_0^{(g)}$ . Hence, we have  $\tilde{g}(X_i) = 1$  and  $\tilde{g}(Y_i) = 0$ , without requiring any additional properties from  $g$ . This is exactly the case if  $f$  is monotone. Therefore, we obtain an analog of [17, Lemma 4]:

**Lemma 6.5** (Direct sum is reducible to  $KW_{f \diamond g}^+$ ). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function, and let  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be a Boolean function. Then, it holds that:*

$$KW_f^+ \otimes KW_g^u \leq KW_{f \diamond g}^u.$$

PROOF. TOPROVE 32 □

We are now ready to prove Proposition 1.37, which we restate to ease the reading.



**Proposition 1.37** (Lower bound via direct sum). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Let  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  a Boolean function. Let  $\Phi : X \times Y \rightarrow \{0, 1\}$  and  $\Psi : X' \times Y' \rightarrow \{0, 1\}$  be functions such that  $\Phi$  reduces to  $KW_f^*$  and  $\Psi$  reduces to  $KW_g^u$ . Then, the following holds:*

$$CC(KW_{f \circ g}^u) \geq \log(\text{Rank}(M_\Phi)) + \log(\text{Rank}(M_\Psi)).$$

PROOF. **TOPROVE 33**

□

Proposition 1.37 provides a step toward advancing hazard-free lower bound techniques, and we hope it encourages further research in this direction. As an application of Proposition 1.37, we prove a lower bound on the hazard-free formula depth of the set covering function composed with the multiplexer function. The set covering function is also used in [17, Theorem 13], where it is composed with itself.

**Definition 6.6** (Set covering  $MC_{k,n}$  [16, Example 5.2.1]). *Given a bipartite graph  $G = (U \cup V, E)$  with  $|U| = |V| = n$  we have  $MC_{k,n}(G) = 1 \iff$  there exists a  $U' \subseteq U$  such that  $|U'| = k$  and every node in  $V$  has a neighbor in  $U'$ .*

It is not hard to see that  $MC_{k,n}$  is monotone (in the edge set of  $G$ ). Moreover, the disjointness function reduces to  $KW_{MC_{k,n}}^*$ .

**Definition 6.7** (The disjointness function). *Let  $\mathbf{P}([n])$  denote all subsets of  $[n]$ , and let  $\mathbf{P}_l([n])$  denote all subsets of  $[n]$  of size  $l$ :*

- Let  $I_n : \mathbf{P}([n]) \times \mathbf{P}([n]) \rightarrow \{0, 1\}$  where  $I_n(S, T) = 0 \iff S \cap T = \emptyset$ .
- For  $l \leq n/2$ , let  $I_{l,n} : \mathbf{P}_l([n]) \times \mathbf{P}_l([n]) \rightarrow \{0, 1\}$  where  $I_{l,n}(S, T) = 0 \iff S \cap T = \emptyset$ .

**THEOREM 6.8** ( $I_{k,n}$  IS REDUCIBLE TO SET COVERING [31], [16, THEOREM 5.2.1]). *Let  $k = c \log n$  for some suitable  $c > 0$ . Then  $I_{k,n} \leq KW_{MC_{k,n}}^*$ .*

[13] proved that the subcube-intersect $_n$  reduces to  $KW_{MUX_n}^u$ .

**Definition 6.9** (subcube-intersect [13, p.15]). *Let subcube-intersect $_n : \{0, u, 1\}^n \times \{0, u, 1\}^n \rightarrow \{0, 1\}$  be a function such that subcube-intersect $(x, y) = 1$  if and only if  $x$  and  $y$  have a common resolution.*

**Lemma 6.10** ([13, Lemma 5.10]). *The subcube-intersect $_n$  communication problem reduces to the communication problem  $KW_{MUX_n}^u$  with no extra cost.*

Since both  $I_{l,n}$  and subcube-intersect $_n$  are functions, the communication complexity of the associated problem can be lower bounded by the rank of the corresponding matrices.

**THEOREM 6.11** ([15]).

$$\text{Rank}(M_{I_n}) = 2^n, \text{Rank}(M_{I_{l,n}}) = \binom{n}{l}.$$

**Lemma 6.12** ([13, Lemma 5.12]). *Rank( $M_{\text{subcube-intersect}_n}$ ) =  $3^n$  for every  $n \geq 1$ .*

**Proposition 6.13** (Composition of the set covering and multiplexer functions). *Let  $MUX_m$  be the multiplexer function as in Definition 1.23. Let  $MC_{k,n}$  be the set covering function as in Definition 6.6, such that  $k = c \log n$  for some suitable constant  $c > 0$  as in Theorem 6.8. Then,*

$$\text{depth}_F^u(MC_{k,n} \diamond MUX_m) \geq \log(3^m) + \log \binom{n}{c \log n} \geq \log(3) \cdot m + \Omega(c(\log n)^2).$$

PROOF. **TOPROVE 34**

□

**Remark 6.14.** As  $MC_{k,n}$  is monotone, it equals its hazard-derivative at  $\bar{0}$ , which is also the “hardest” derivative (this can be inferred from the proof of Theorem 1.12). Hence, a formula for the hazard-derivative of  $MC_{k,n} \diamond MUX_m$  can be obtained from composing the monotone formula for  $MC_{k,n}$  with that of the hazard-derivative of  $MUX_m$  (see the “chain rule” in [12, Lemma 4.7]). As shown in [13, Proposition 1], every hazard-derivative of  $MUX_m$ , has a formula of depth  $m + \lceil \log(m+1) \rceil$ . Thus, every hazard-derivative of  $MC_{k,n} \diamond MUX_m$  has a monotone formula of depth  $\lceil \log \binom{n}{k} \rceil + \lceil \log n \rceil + \lceil \log k \rceil + m + \lceil \log(m+1) \rceil$ , which for our setting of parameters would give a lower bounds of  $m + (1 + o(1))c \log^2 n$ , which is worse than what Proposition 6.13 yields.

**Remark 6.15.** Since we are using a monotone outer function, the advantage that we can get compared to the hazard-derivative method must come from the inner function.

## 7 Open problems

The most obvious open problem is determining whether the universal hazard-free formula size upper bound in the worst case is equal to that of a random function.

**Question 7.1.** Is it true that for every Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ ,  $\text{size}_F^u(f) \leq 2^{(1+o(1))n}$ ?

Another interesting question is whether Theorem 1.31 also holds for circuits:

**Question 7.2.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a non-constant Boolean function. Is it true that there exists  $x \in \{0, 1\}^n$  such that:

$$\text{size}_C^+(df(x; \cdot)) = \text{size}_C^u(f),$$

if and only if  $f$  is a unate function?

The validity of the KRW conjecture in the hazard-free setting is also an intriguing question. Many previous works on the KRW conjecture have considered restricted versions to gain intuition and develop tools, including [5, 8, 10, 17, 21, 25], among others. We believe that the hazard-free KRW conjecture presents an interesting restricted model to study, with potential applications to all Boolean functions.

**Conjecture 1.28** (Hazard-Free KRW conjecture). Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^m \rightarrow \{0, 1\}$  be non-constant Boolean functions. Then the following holds:

$$\text{depth}_F^u(f \diamond g) \approx \text{depth}_F^u(f) + \text{depth}_F^u(g).$$

Finally, we consider block compositions with the XOR function. Recall that the hazard-free complexity of  $\text{XOR}_n$  is equal to its standard formula complexity. Additionally, its prime implicants and prime impicates are  $f^{-1}(1)$  and  $f^{-1}(0)$ , respectively, meaning its hazard-free and the classic KW games are equivalent. This naturally leads to the question of whether the lower bound proof of Dinur and Meir [4] can be adapted to the hazard-free setting. Specifically, we wonder if, in the following theorem, one can replace  $\text{size}_F$  with  $\text{size}_F^u$ .

**THEOREM 7.3** (COMPOSITION OVER PARITY [4, THEOREM 3.1]). Let  $f : \{0, 1\}^k \rightarrow \{0, 1\}$  be a non-constant Boolean function. Then:

$$\text{size}_F(f \diamond \text{XOR}_m) \geq \frac{\text{size}_F(f) \cdot \text{size}_F(\text{XOR}_m)}{2^{\tilde{O}(\sqrt{k+\log m})}},$$

where  $\tilde{O}(t) := O(t \cdot \log^{O(1)} t)$ .

## Acknowledgments

The first author would like to thank Maya Baruch for her invaluable discussions on various topics, insightful feedback on the proofs, and support in articulating ideas clearly. She is also grateful to Ido Weinstein and Rotem Zluf for their willingness to listen to her early ideas, which contributed to identifying and correcting mistakes in the initial stages of this project.

## References

- [1] Noga Alon and Ravi B. Boppana. 1987. The monotone circuit complexity of Boolean functions. *Combinatorica* 7 (1987), 1–22. doi:10.1007/BF02579196
- [2] Alexander E. Andreev. 1987. On a method for obtaining more than quadratic effective lower bounds for the complexity of  $\pi$ -schemes. *Moscow Univ. Math. Bull.* 42, 1 (1987), 63–66.
- [3] Johannes Bund, Christoph Lenzen, and Moti Medina. 2020. Optimal Metastability-Containing Sorting via Parallel Prefix Computation. *IEEE Trans. Comput.* 69, 2 (2020), 198–211. doi:10.1109/TC.2019.2939818
- [4] Irit Dinur and Or Meir. 2018. Toward the KRW Composition Conjecture: Cubic Formula Lower Bounds via Communication Complexity. *Computational Complexity* 27 (September 2018), 375–462. doi:10.1007/s00037-017-0159-x
- [5] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiří Sgall. 2001. Communication complexity towards lower bounds on circuit depth. *Comput. Complexity* 10, 3 (2001), 210–246. doi:10.1007/s00037-001-8195-x
- [6] Edward B. Eichelberger. 1965. Hazard detection in combinational and sequential switching circuits. *IBM Journal of Research and Development* 9, 2 (1965), 90–99. doi:10.1147/rd.92.0090
- [7] Stephan Friedrichs, Matthias Függer, and Christoph Lenzen. 2018. Metastability-Containing Circuits. *IEEE Trans. Comput.* 67, 8 (2018), 1167–1183. doi:10.1109/TC.2018.2808185
- [8] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. 2017. Toward Better Formula Lower Bounds: The Composition of a Function and a Universal Relation. *SIAM J. Comput.* 46, 1 (2017), 114–131. doi:10.1137/15M1018319
- [9] Johan Håstad. 1998. The shrinkage exponent of de Morgan formulas is 2. *SIAM J. Comput.* 27, 1 (1998), 48–64. doi:10.1137/S0097539794261556
- [10] Johan Håstad and Avi Wigderson. 1993. Composition of the universal relation. In *Advances in computational complexity theory (New Brunswick, NJ, 1990)*. DIMACS Ser. Discrete Math. Theoret. Comput. Sci., Vol. 13. Amer. Math. Soc., Providence, RI, 119–134. doi:10.1090/dimacs/013/07
- [11] David A. Huffman. 1957. The Design and Use of Hazard-Free Switching Networks. *J. ACM* 4, 1 (January 1957), 47–62. doi:10.1145/320856.320866
- [12] Christian Ikenmeyer, Balagopal Komarath, Christoph Lenzen, Vladimir Lysikov, Andrey Mokhov, and Kartteek Sreenivasaiiah. 2019. On the Complexity of Hazard-free Circuits. *J. ACM* 66, 4, Article 25 (Aug. 2019), 20 pages. doi:10.1145/3320123
- [13] Christian Ikenmeyer, Balagopal Komarath, and Nitin Saurabh. 2023. Karchmer-Wigderson Games for Hazard-Free Computation. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 251)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Oktavie-Allee, 66687 Wadern, Germany, 74:1–74:25. doi:10.4230/LIPIcs.ITCS.2023.74
- [14] Stasys Jukna. 2021. Notes on Hazard-Free Circuits. *SIAM Journal on Discrete Mathematics* 35, 2 (2021), 770–787. doi:10.1137/20M1355240
- [15] William M. Kantor. 1972. On incidence matrices of finite projective and affine spaces. *Mathematische Zeitschrift* 124 (1972), 315–318. doi:10.1007/BF01113923
- [16] Mauricio Karchmer. 1989. *Communication Complexity: A New Approach to Circuit Depth*. The MIT Press, Cambridge, MA, USA. doi:10.7551/mitpress/1948.001.0001
- [17] Mauricio Karchmer, Ran Raz, and Avi Wigderson. 1995. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity* 5 (1995), 191–204. doi:10.1007/BF01206317
- [18] Mauricio Karchmer and Avi Wigderson. 1990. Monotone Circuits for Connectivity Require Super-Logarithmic Depth. *SIAM Journal on Discrete Mathematics* 3, 2 (1990), 255–265. doi:10.1137/0403021
- [19] Stephen Cole Kleene. 1952. *Introduction to metamathematics*. D. Van Nostrand Co., Inc., New York, NY, USA. x+550 pages.
- [20] Balagopal Komarath and Nitin Saurabh. 2020. On the complexity of detecting hazards. *Inf. Process. Lett.* 162 (2020), 105980. doi:10.1016/j.ipl.2020.105980
- [21] Sajin Koroth and Or Meir. 2018. Improved Composition Theorems for Functions and Relations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20–22, 2018 - Princeton, NJ, USA (LIPIcs, Vol. 116)*, Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Oktavie-Allee, 66687 Wadern, Germany, 48:1–48:18. doi:10.4230/LIPIcs.APPROX-

RANDOM.2018.48

- [22] Eyal Kushilevitz and Noam Nisan. 1996. *Communication Complexity*. Cambridge University Press, Cambridge, MA, USA. doi:10.1017/CBO9780511574948
- [23] EJ McCluskey. 1962. Transients in combinational logic circuits. *Redundancy Technique for Computing System* 9 (1962), 9–46.
- [24] Edward J. McCluskey. 1986. *Logic design principles - with emphasis on testable semicustom circuits*. Prentice Hall, Saddle River, New Jersey, USA.
- [25] Or Meir. 2020. Toward Better Depth Lower Bounds: Two Results on the Multiplexor Relation. *Computational Complexity* 29, 1 (2020), 4. doi:10.1007/s00037-020-00194-8
- [26] Masao Mukaidono. 1972. On the B-ternary logical function—A ternary logic considering ambiguity. *Syst. Comput. Controls* 3, 3 (1972), 27–36.
- [27] David E. Muller. 1956. Complexity in electronic switching circuits. *IRE Transactions on Electronic Computers* EC-5, 1 (1956), 15–19. doi:10.1109/TEC.1956.5219786
- [28] Steven M. Nowick and David L. Dill. 1995. Exact two-level minimization of hazard-free logic with multiple-input changes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 14, 8 (1995), 986–997. doi:10.1109/43.402498
- [29] Anup Rao and Amir Yehudayoff. 2020. *Communication Complexity: and Applications*. Cambridge University Press, Cambridge, MA, USA. doi:10.1017/9781108671644
- [30] Alexander A. Razborov. 1985. Lower bounds on the monotone complexity of some Boolean function. *Soviet Math. Dokl.* 31 (1985), 354–357.
- [31] Alexander A. Razborov. 1990. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica* 10 (1990), 81–93. doi:10.1007/BF02122698
- [32] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. 2024. KRW Composition Theorems via Lifting. *Computational Complexity* 33, 1 (2024), 1–97. doi:10.1007/s00037-024-00250-7
- [33] John Riordan and Claude E. Shannon. 1942. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics* 21, 1-4 (1942), 83–93. doi:10.1002/sapm194221183
- [34] Claude E. Shannon. 1949. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal* 28, 1 (1949), 59–98. doi:10.1002/j.1538-7305.1949.tb03624.x
- [35] Avishay Tal. 2014. Shrinkage of De Morgan Formulae by Spectral Techniques. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1730 Massachusetts Ave., NW Washington, DC, United States, 551–560. doi:10.1109/FOCS.2014.65
- [36] Éva Tardos. 1988. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica* 8 (1988), 141–142. doi:10.1007/BF02122563
- [37] S. Unger. 1959. Hazards and Delays in Asynchronous Sequential Switching Circuits. *IRE Transactions on Circuit Theory* 6, 1 (1959), 12–25. doi:10.1109/TCT.1959.1086527
- [38] Michael Yoeli and Shlomo Rinon. 1964. Application of Ternary Algebra to the Study of Static Hazards. *J. ACM* 11, 1 (Jan. 1964), 84–97. doi:10.1145/321203.321214