On the Complexity of Hazard-Free Formulas

Leah London Arazi
School of Computer Science, Tel Aviv University, Israel

Amir Shpilka
School of Computer Science, Tel Aviv University, Israel

Abstract

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

Our first result shows that unate 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.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Communication complexity

Keywords and phrases Hazard-free computation, Boolean formulas, monotone formulas, Karchmer-Wigderson games, communication complexity, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.115

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2411.09026` [2]

Funding 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.

Acknowledgements The first author thanks Maya Baruch for invaluable discussions and feedback. She also thanks Ido Weinstein and Rotem Zluf for discussing early ideas, which helped identify and correct initial mistakes.

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis
Article No. 115; pp. 115:1–115:19

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 [10], 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 [30, 20]. There is a broad body of literature on hazard-free computation, and we refer the interested reader to [19, 11] and references therein.

Recent research on hazard-free computation focuses on two main areas: electronic circuit design (e.g., [7, 3]) and computational complexity (e.g., [11, 17, 13, 12]). While the motivation for the former is clear, the interest from the computational complexity community stems from the results of [11], 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 [31, 6]. This logic extends the standard Boolean values $\{0, 1\}$ by introducing a third symbol, $\mathfrak{u}$, representing an undefined or unstable value.

▶ **Definition 1** (Kleene three-valued logic [16]). *Kleene's three-valued strong logic of indeterminacy extends the two-valued Boolean logic by a third value* $\mathfrak{u}$. *The Boolean values* $\{0, 1\}$ *are called* "stable" *and* $\mathfrak{u}$ *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 $x \in \{0, \mathfrak{u}, 1\}^n$ and outputting a value in $\{0, \mathfrak{u}, 1\}$. The extended gates are monotone with respect to the following partial order:

▶ **Definition 2** ($\overset{\mathfrak{u}}{\leq}$ [22]). *The relation* $\overset{\mathfrak{u}}{\leq}$ *is a* partial *order in which* $\mathfrak{u} \overset{\mathfrak{u}}{\leq} 1$, $\mathfrak{u} \overset{\mathfrak{u}}{\leq} 0$ *and* $0, 1$ *are not comparable. For* $x, y \in \{0, \mathfrak{u}, 1\}^n$, *we denote* $x \overset{\mathfrak{u}}{\leq} y$ *if for every* $i \in [n]$, $x_i \overset{\mathfrak{u}}{\leq} y_i$.

We present an abstraction of hazards using ternary algebra, following the definitions of [7, 11].

Table 1 Extended De Morgan gates.

or	0	$\mathfrak{u}$	1
0	0	$\mathfrak{u}$	1
$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	1
1	1	1	1

and	0	$\mathfrak{u}$	1
0	0	0	0
$\mathfrak{u}$	0	$\mathfrak{u}$	$\mathfrak{u}$
1	0	$\mathfrak{u}$	1

not	0	$\mathfrak{u}$	1
	1	$\mathfrak{u}$	0

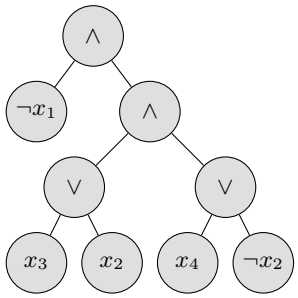


Figure 1 Formula with a hazard at $x_1 = \mathfrak{u}$, $x_2 = \mathfrak{u}$, $x_3 = 0$, $x_4 = 0$.

▶ **Definition 3** (Resolution). *Let $x \in \{0, \mathfrak{u}, 1\}^n$. A resolution $y \in \{0,1\}^n$ of x is obtained by replacing every occurrence of $\mathfrak{u}$ in x by either 0 or 1. We denote by $R(x)$ the set of all resolutions of x, namely, $R(x) := \{y \in \{0,1\}^n \mid x \overset{\mathfrak{u}}{\leq} y\}$.*

▶ **Definition 4** (Hazard). *Let C be a Boolean circuit. C has a* hazard *at input $x \in \{0, \mathfrak{u}, 1\}^n$ if and only if $C(x) = \mathfrak{u}$ 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 $x \in \{0, \mathfrak{u}, 1\}^n$ 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 $\mathfrak{u}$. Put differently, a hazard-free circuit outputs $\mathfrak{u}$ if and only if the output cannot be determined solely by the stable bits.

▶ **Definition 5** (Hazard-free circuit). *Let C be a Boolean circuit. C is said to be* hazard-free *if for every $x \in \{0, \mathfrak{u}, 1\}^n$, 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[1] 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 constant Boolean function 1, and by the truth tables in Table 1, has a hazard at $x = \mathfrak{u}$. 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 $R(\mathfrak{uu}00)$ the formula outputs 0, but on $\mathfrak{uu}00$, the formula outputs $\mathfrak{u}$.

Not every ternary function $f : \{0, \mathfrak{u}, 1\}^n \to \{0, \mathfrak{u}, 1\}$ 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 $\overset{\mathfrak{u}}{\leq}$. 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. [22] refers to these functions as *natural functions*.

▶ **Definition 6** (Natural function). *A function $f : \{0, \mathfrak{u}, 1\}^n \to \{0, \mathfrak{u}, 1\}$ 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 $\overset{\mathfrak{u}}{\leq}$. Intuitively, replacing stable bits in the input with $\mathfrak{u}$ can only cause the output to change to $\mathfrak{u}$.*

[1] An example of a Boolean circuit exhibiting hazards is shown in [11, Figure 1].

▶ **Proposition 7** ([22, Theorem 3]). *A function $f : \{0,\mathrm{u},1\}^n \to \{0,\mathrm{u},1\}$ 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 8** (The hazard-free extension of f). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. The* hazard-free extension of f, *denoted $\tilde{f} : \{0,\mathrm{u},1\}^n \to \{0,\mathrm{u},1\}$, is defined as follows:*

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

[11] 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 9** (Hazard-free and monotone circuit complexity are equivalent [11, Theorem 1.3]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function, then:*

$$size_C^{\mathrm{u}}(f) = size_C^{+}(f),$$

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

In fact, [13] further proved that an optimal hazard-free circuit for a monotone function must be a monotone circuit. As superpolynomial lower bounds for monotone circuits [25, 1, 29] are known, [11] used Theorem 9 to prove lower bounds on the hazard-free circuit complexity of monotone functions whose Boolean circuit complexity is only polynomial.

While Theorem 9 only concerns monotone functions, [11] 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 $f : \{0,1\}^n \to \{0,1\}$ is bounded from below by the monotone complexity of the *hazard-derivative* of f, defined as follows:

▶ **Definition 10** (The hazard-derivative of a natural function [11, Definition 4.3]). *Let $f : \{0,\mathrm{u},1\}^n \to \{0,\mathrm{u},1\}$ be a natural ternary function. The* hazard-derivative *of f, denoted $\mathrm{d}f : \{0,1\}^{2n} \to \{0,1\}$, is defined as follows:*

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

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

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

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

The + operator in Definition 10 also serves as the hazard-free XOR operator.[2]

For a fixed $x \in \{0,1\}^n$, $\mathrm{d}f(x;y)$ indicates whether "perturbing" x according to $\mathfrak{u}y$ causes $\tilde{f}$ to output an unstable value. It is not difficult to see that $\mathrm{d}f(x;\cdot)$ 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 11** (The hazard-derivative is monotone [11, Lemma 4.6]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and $x \in \{0,1\}^n$ be a fixed input, then $\mathrm{d}f(x;\cdot)$ is a* monotone *Boolean function.*

Every hazard-free circuit (or formula) for f can be tweaked to yield a monotone circuit (or formula) for $\mathrm{d}f(x;\cdot)$ resulting in the following Theorem:[3]

▶ **Theorem 12** (The hazard-derivative lower bound [11, Theorem 4.9]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function and $x \in \{0,1\}^n$ be a fixed Boolean input, then:*

$$size^{+}_{C}(\mathrm{d}f(x;\cdot)) \leq size^{\mathfrak{u}}_{C}(f) \quad \text{and} \quad size^{+}_{F}(\mathrm{d}f(x;\cdot)) \leq size^{\mathfrak{u}}_{F}(f),$$

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

[11], and later [13], 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 [11], hazard-free lower bounds were proved only for restricted models such as hazard-free depth 2 circuits [6, 24]. Therefore, their work motivates the study of the hazard-free model as a bridge between monotone complexity and Boolean circuit complexity.

Additionally, [11, Theorem 1.3] proved that the hazard-derivative method yields *tight* bounds for monotone Boolean functions. That is, for a monotone function $\mathrm{size}^{+}_{C}(\mathrm{d}f(x;\cdot)) = \mathrm{size}^{\mathfrak{u}}_{C}(f)$ and $\mathrm{size}^{+}_{F}(\mathrm{d}f(x;\cdot)) = \mathrm{size}^{\mathfrak{u}}_{F}(f)$ when $x = \bar{0}$. This raises the following intriguing question.

▶ **Question 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

[11, 13] obtained exponential lower bounds on the hazard-free computation of many Boolean functions using Theorem 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. [12] 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, [12] note that the hazard-derivatives of the parity function on n variables, 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 XOR_n is $\Omega(n^2)$, while it is not hard to see that any formula for XOR_n is hazard-free.

---

[2] When considering $\mathfrak{u}y_i = \begin{cases} 0, & \text{if } y_i = 0, \\ \mathfrak{u}, & \text{if } y_i = 1. \end{cases}$

[3] [11, Theorem 4.9] only speaks about circuit size, but the same proof holds for formulas as well.

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 to the one defined in [13, Section 7], which we refer to as the *monotone gap*.

▶ **Definition 14** (The monotone gap). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. The* monotone gap of f *is defined as:*

$$\textit{mono-gap}(f) := \frac{\textit{size}^{\mathfrak{u}}_F(f)}{\max\limits_{x\in\{0,1\}^n} \textit{size}^{+}_F(\mathrm{d}f(x;\cdot))}.$$

For example, the monotone gap of XOR_n is $\Omega(n)$. In an attempt to understand the power of the hazard-derivative method (or its weaknesses), we wish to answer the following:

▶ **Question 15.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. How large can mono-gap(f) be? What is mono-gap(f) for a typical Boolean function?*

1.2 The Hazard-Free KW Game

[12] studied a generalization of the classic Karchmer-Wigderson game [15] in the hazard-free setting.

▶ **Definition 16** (Hazard-free KW game [12, Definition 5]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. The* hazard-free Karchmer-Wigderson game *of f, denoted $KW^{\mathfrak{u}}_f$, is defined as follows: Alice receives $x \in \{0,\mathfrak{u},1\}^n$ such that $\tilde{f}(x) = 1$, Bob receives $y \in \{0,\mathfrak{u},1\}^n$ 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 $x_i, y_i \neq \mathfrak{u}$.*

As $\mathrm{KW}^{\mathfrak{u}}_f \subseteq \tilde{f}^{-1}(1) \times \tilde{f}^{-1}(0) \times [n]$, and every $x \in \tilde{f}^{-1}(1)$ and $y \in \tilde{f}^{-1}(0)$ are valid inputs[4], $\mathrm{KW}^{\mathfrak{u}}_f$ 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 17** ([12, Theorem 7]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Then,*

$$\textit{depth}^{\mathfrak{u}}_F(f) = CC(KW^{\mathfrak{u}}_f) \textit{ and } \textit{size}^{\mathfrak{u}}_F(f) = \textit{mono-rec}(KW^{\mathfrak{u}}_f),$$

where $\textit{depth}^{\mathfrak{u}}_F(f)$ is the depth of the shallowest hazard-free formula computing f, $CC(\mathcal{R})$ is the communication complexity of the relation $\mathcal{R}$, and 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, [12] proved that the communication complexity of $\mathrm{KW}^{\mathfrak{u}}_f$ does not decrease even if it is played only on the *prime implicants* of f and the *prime implicates* of f.

▶ **Theorem 18** ([12, Theorem 10]). *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$, the communication complexity of $KW^{\mathfrak{u}}_f$ remains unchanged even if we restrict Alice's input to the prime implicants of f and Bob's input to the prime implicates of f.*

The notion of prime implicants and prime implicates is well established in Boolean algebra. Following [12], we provide analogous definitions using ternary algebra as previously defined in [17][p.3] and [12][p.7].

[4] 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 8.

▶ **Definition 19** (Implicants and implicates of a Boolean function). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We define the implicants of f, denoted by $I_1^{(f)}$, and the implicates 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 20** (Prime implicants and prime implicates of a Boolean function). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We say $x \in \{0,\mathfrak{u},1\}^n$ 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 $\tilde{f}(x|_{i \leftsquigarrow \mathfrak{u}}) = \mathfrak{u}$, where $x|_{i \leftsquigarrow b}$ stands for replacing the value of the i'th coordinate in x with $b \in \{0,\mathfrak{u},1\}$.*

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

For the remainder of the paper, we abuse notation and use $\mathrm{KW}_f^{\mathfrak{u}}$ to denote the restricted game. We next define the communication matrix of the $\mathrm{KW}_f^{\mathfrak{u}}$ game.

▶ **Definition 21** (Communication matrix of $\mathrm{KW}_f^{\mathfrak{u}}$). *Let $\mathcal{X} := (p_1^{(1)}, \ldots, p_1^{(k)})$ and $\mathcal{Y} := (p_0^{(1)}, \ldots, p_0^{(\ell)})$ 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 $KW_f^{\mathfrak{u}}$, denoted $M_{KW_f^{\mathfrak{u}}}$, is defined such that every entry of $M_{KW_f^{\mathfrak{u}}}$ contains the set of all stable coordinates in which $p_1^{(i)}$ and $p_0^{(j)}$ differ:*

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

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

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

▶ **Theorem 22** ([12, Theorem 11]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function. Then, $KW_f^{\mathfrak{u}}$ 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 9 for formulas instead of circuits.

By using communication complexity methods to prove lower bounds on hazard-free formulas, [12, Theorems 19, 23] determined the exact hazard-free formula size of the multiplexer function.

▶ **Definition 23** (The multiplexer function [12]). *The function $MUX_n : \{0,1\}^{n+2^n} \to \{0,1\}$, called the multiplexer function, is defined by*

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

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

▶ **Theorem 24** (Hazard-free formula size for the multiplexer function [12, Exact Bounds]).

$$size_F^{\mathfrak{u}}(MUX_n) = 2 \cdot 3^n - 1.$$

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 [12, Proposition 1]).

1.3 Universal Upper Bounds

An immediate consequence of Theorem 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:

$$f(x) = \mathrm{MUX}_n(x, f(0,0,\ldots,0), \ldots, f(1,1,\ldots,1)),$$

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 [10].

We note that this state of knowledge is much poorer compared to what is known for hazard-free circuits. Analogous to [28, 23], Jukna [13, Section 7] proved that any Boolean function has a hazard-free circuit of size $O(2^n/n)$, applying the following:

▶ **Proposition 25** ([13, Proposition 3]). *Let C_0, C_1 be arbitrary hazard-free circuits over $x_1, \ldots, x_{n-1}$, and x_n a new variable. Then, the circuit*

$$C := (\neg x_n \wedge C_0) \vee (x_n \wedge C_1) \vee (C_0 \wedge C_1) \tag{1}$$

is hazard-free.

Therefore, naturally, we can construct a hazard-free circuit for $f : \{0,1\}^n \to \{0,1\}$ by using hazard-free implementations C_0 and C_1 for $f_0(x_1, \ldots, x_{n-1}) := f(x_1, \ldots, x_{n-1}, 0)$ and $f_1(x_1, \ldots, x_{n-1}) := f(x_1, \ldots, x_{n-1}, 1)$, 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.[5]

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 $\mathrm{size}^{\mathfrak{u}}_C(f) = \Theta(\mathrm{size}_C(f))$.

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 [27, 28], whereas Theorem 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 26.** *Denote with $\mathrm{size}^{\mathfrak{u}}_F(n)$ the maximal hazard-free complexity of an n-bit Boolean function. What is the asymptotic value of $\mathrm{size}^{\mathfrak{u}}_F(n)$? What is $\mathrm{size}^{\mathfrak{u}}_F(f)$ of a typical Boolean function?*

1.4 The Hazard-Free KRW Conjecture

Karchmer, Raz and Wigderson [14] introduced the KRW conjecture as an approach for establishing super-polynomial lower bounds for De Morgan formulas. Proving this conjecture would imply $\mathrm{P} \not\subseteq \mathrm{NC}^1$. A detailed discussion can be found in [21].

[5] Note that the extra $(C_0 \wedge C_1)$ term is necessary. Say that for some $x \in \{0, \mathfrak{u}, 1\}^{n-1}$ we have $\tilde{f}(x, \mathfrak{u}) = 1$. 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 $\mathfrak{u}$.

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

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

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 27** (The KRW conjecture [14]). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be non-constant Boolean functions. Then it holds that:*[6]

$$depth_F(f \diamond g) \approx depth_F(f) + depth_F(g).$$

Following [15], the conjecture can be rephrased as:

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

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

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

Similarly to [4, 26], we state a stronger version of the conjecture on the formula size.

▶ **Conjecture 29** (Hazard-free strong KRW conjecture). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be non-constant Boolean functions. Then the following holds:*

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

The KRW conjecture has been extensively studied through various special cases and simplified communication problems, as intermediate steps toward proving Conjecture 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.

▶ **Conjecture 30** (Monotone KRW conjecture [14]). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be monotone Boolean functions. Then the following holds:*

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

In [26, Theorem 3.1], Conjecture 30 was proved for monotone inner functions g whose depth complexity can be lower-bounded by a lifting theorem. In an earlier work, [14] 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 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.

[6] We use $\approx$ instead of $\geq$ as we can allow a less exact inequality, see [14, 8].

1.5 Our Results

1.5.1 Unateness as a Criterion

We provide a complete answer to Question 13 by proving that the hazard-derivative method yields exact lower bounds only for unate Boolean functions:

▶ **Theorem 31** (Unateness as a criterion). *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant Boolean function. Then, there exists $x \in \{0,1\}^n$ such that:*

$$size_F^{+}(\mathrm{d}f(x;\cdot)) = size_F^{\mathrm{u}}(f),$$

if and only if f is a unate function.

A unate Boolean function is defined as follows:

▶ **Definition 32** (Unate function in x_i). *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is* positive unate *in x_i if for all $x \in \{0,1\}^n$ the following holds:*

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

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

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

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

▶ **Definition 33** (Unate function). *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is* unate *if it is unate in $x_1, \ldots, x_n$.*

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. See Section 2 for more details.

1.5.2 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 26. This gives an exponential improvement over the best known $O(3^n)$ upper bound of [12]. It is important to note that the upper bound of [12] 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 34** (A universal upper bound for most Boolean functions). *For a random Boolean function $f : \{0,1\}^n \to \{0,1\}$, we get that with high probability:*

$$\frac{2^n}{\log n} \leq size_F^{\mathrm{u}}(f) \leq 2^n \cdot n^{(1-o(1))\log n}.$$

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 34. The second proof 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 15, showing that the monotone gap can be exponential in the number of variables.

▶ **Proposition 35.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a random Boolean function. Then, with high probability,*

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

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 $p_b \overset{u}{\leq} x$. We denote the subset of $P_b^{(f)}$ containing the strings that derive x by $P_b^{(f)}|_x$.

▶ **Theorem 36** (Hazard-derivatives upper bound)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ and $b \in \{0,1\}$. Let $X_b \subseteq f^{-1}(b)$. Then,*

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

We also provide an example in Section 2.3 that shows that Theorem 36 yields tight bounds for range functions (functions that accept all inputs whose weight is within some range).

1.5.3 Composition of Hazard-Free Functions

Following [14], 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 37** (Lower bound via direct sum)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function. Let $g : \{0,1\}^m \to \{0,1\}$ a Boolean function. Let $\Phi : X \times Y \to \{0,1\}$ and $\Psi : X' \times Y' \to \{0,1\}$ be* functions *such that Φ reduces to KW_f^+ and Ψ reduces to KW_g^u. Then, the following holds:*

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

Proposition 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.

▶ **Proposition 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,*

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

2 Proof Overview

We provide a brief overview of the proofs, focusing on the main ideas and the underlying intuition. For the full proofs, see [2].

	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_{prime}.

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

(b) The matrix of $\mathrm{d}f_{\text{prime}}(1100;\cdot)$.

	u00u
u1uu	2
uu1u	3

(c) The matrix of $\mathrm{d}f_{\text{prime}}(1001;\cdot)$.

	000u
uu1u	3
u1uu	2
1uuu	1

(d) The matrix of $\mathrm{d}f_{\text{prime}}(1111;\cdot)$.

Figure 2 Sketch of KW^u communication matrices for f_{prime}.

Our proofs rely on the framework of hazard-free KW games, restricted to the prime implicants and prime implicates (see Theorem 18). Specifically, we analyze the hazard-free KW game of the hazard-derivative at a fixed Boolean input and its relation to that of the original Boolean function. First, we demonstrate how to derive the prime implicates and prime implicants of the hazard-derivative from those of the function itself. We then show that the communication matrix of the hazard-derivative is a submatrix of the original function's communication matrix.

We present a simple example which will serve as an illustration throughout this section. Consider $f_{\text{prime}} : \{0,1\}^4 \to \{0,1\}$, 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 implicates of f_{prime} and sketch its hazard-free KW game communication matrix, as demonstrated in Figure 2a.

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and $x \in \{0,1\}^n$ a fixed Boolean input such that $f(x) = b \in \{0,1\}$. We begin by presenting the connection between the prime implicants and prime implicates of f and $\mathrm{d}f(x;\cdot)$.

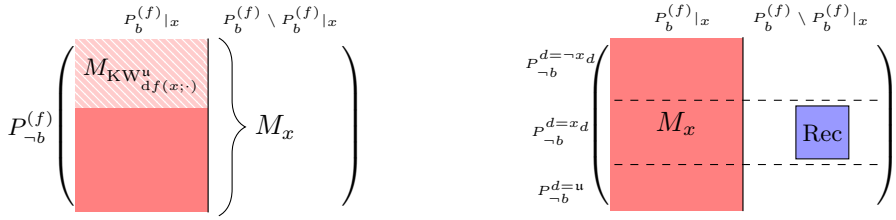
▶ **Lemma 39** (The prime implicates of the hazard-derivative)**.**

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

▶ **Lemma 40** (The prime implicants of the hazard-derivative)**.**

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

To demonstrate the claims consider Figure 2b. For the Boolean input 1100, we have $f_{\text{prime}}(1100) = 0$. It can be verified that $P_0^{(f_{\text{prime}})}|_{1100} = \{\mathrm{uu00}, \mathrm{u1u0}, \mathrm{1uu0}\}$. By Lemma 39, the prime implicates of $\mathrm{d}f_{\text{prime}}(1100;\cdot)$ are $\{\mathrm{uu00} + 1100, \mathrm{u1u0} + 1100, \mathrm{1uu0} + 1100\} = \{\mathrm{uu00}, \mathrm{u0u0}, \mathrm{0uu0}\}$. 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 only two (or less) out of these three bits still preserves the function's value. Following Lemma 40, calculating $(p_1 + 1100)|_{0 \Leftarrow \mathrm{u}}$ for every $p_1 \in P_1^{(f_{\text{prime}})}$ yields $\{\mathrm{111u}, \mathrm{u111}, \mathrm{1u11}, \mathrm{1uu1}, \mathrm{uuu1}\}$. This set does not consist solely of *prime* implicants, as $\mathrm{u111}, \mathrm{1u11}, \mathrm{1uu1}$ are already "covered" by $\mathrm{uuu1}$. Furthermore, this is a subset of $I_1^{(\mathrm{d}f_{\text{prime}}(1100;\cdot))}$ as $y \in R(\mathrm{uuu1})$ perturbs (at least) the least significant bit of 1100, and



(a) $M_{\mathrm{KW}^{\mathtt{u}}_f}$, M_x and $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$. **(b)** $M_{\mathrm{KW}^{\mathtt{u}}_f}$ for a non-unate f.

Figure 3 Sketch of $M_{\mathrm{KW}^{\mathtt{u}}_f}$.

$R(110\mathtt{u})$ contains 1101, which is the prime number 13 (and clearly we can get a non-prime number as well). Similarly, $y \in R(111\mathtt{u})$ perturbs (at least) the three most significant bits, and $R(\mathtt{uuu}0)$ contains 0010, which is the prime number 2.

2.1 Proof Overview of Lemmata 39 and 40

Let us discuss the underlying intuition. The hazard-derivative is a monotone function and so its prime implicants and prime implicates are more structured, namely, $P_1^{(\mathrm{d}f(x;\cdot))} \subseteq \{1,\mathtt{u}\}^n$ and $P_0^{(\mathrm{d}f(x;\cdot))} \subseteq \{0,\mathtt{u}\}^n$. Therefore, it is enough to consider the maximal and the minimal resolutions with respect to $\leq$:

▶ **Definition 41** (The maximal and the minimal resolutions w.r.t $\leq$)**.** *Let $x \in \{0,\mathtt{u},1\}^n$. We call $\hat{x} := x|_{\mathtt{u}\Leftarrow 1}$ the* maximal *resolution of x with respect to $\leq$ and $\check{x} := x|_{\mathtt{u}\Leftarrow 0}$ the* minimal *resolution of x with respect to $\leq$.*

Consider $q_0 \in P_0^{(\mathrm{d}f(x;\cdot))}$. The 1s in $\hat{q_0}$ represent all the coordinates in x that *can* be perturbed while the output $\tilde{f}(x+\mathtt{u}\hat{q_0})$ still equals $f(x)=b$. Since q_0 is prime, it should not be possible to add more $\mathtt{u}$s to it, and consequently to $x+\mathtt{u}\hat{q_0}$. This is the same as saying that $x+\mathtt{u}\hat{q_0} = p_b \in P_b^{(f)}|_x$. Since $\mathtt{u}\hat{q_0} = q_0$, 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 $q_1 \in P_1^{(\mathrm{d}f(x;\cdot))}$. The 1s in $\check{q_1}$ represent all the coordinates in x that *must* be perturbed in order for $\tilde{f}(x+\mathtt{u}\check{q_1})$ to be equal $\mathtt{u}$. Hence, there must be a resolution $z \in R(x+\mathtt{u}\check{q_1})$ for which $f(z) \neq f(x)$. We prove that for every $q_1 \in P_1^{(\mathrm{d}f(x;\cdot))}$ such z is unique among $R(x+\mathtt{u}\check{q_1})$. 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 $p_{\neg b} \overset{\mathtt{u}}{\leq} z$. We then show that $q_1 = (p_{\neg b}+x)|_{0\Leftarrow\mathtt{u}}$, relying on the fact that z is unique.

Note that unlike Lemma 39, 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 $p := (p_{\neg b}+x)|_{0\Leftarrow\mathtt{u}}$ there are multiple $z \in R(x+\mathtt{u}\check{p})$ such that $f(z) \neq f(x)$. For those, we prove that $(p_{\neg b}+x)|_{0\Leftarrow\mathtt{u}}$ are implicants of $\mathrm{d}f(x;\cdot)$ (but not prime).

Let $M_{\mathrm{KW}^{\mathtt{u}}_f}$ and $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$ represent the communication matrix of the hazard-free KW game of f and $\mathrm{d}f(x;\cdot)$, respectively. Let M_x be the submatrix of $M_{\mathrm{KW}^{\mathtt{u}}_f}$ with labels $P_{\neg b}^{(f)} \times P_b^{(f)}|_x$. Our characterization of the prime implicants and prime implicates of $\mathrm{d}f(x;\cdot)$, allows us to prove the following, as illustrated in Figure 3a:

▶ **Proposition 42.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant Boolean function. For every $x \in \{0,1\}^n$, $M_{KW^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$ is a submatrix of $M_{KW^{\mathtt{u}}_f}$.*

Proposition 42 can be viewed as an alternative proof for Theorem 12 for formulas. We achieve Proposition 42 through the following observation:

▶ **Lemma 43.** *Let $x \in \{0,1\}^n$ such that $f(x) = b \in \{0,1\}$. Let $q_1 \in P_1^{(\mathrm{d}f(x;\cdot))}$ and $q_0 \in P_0^{(\mathrm{d}f(x;\cdot))}$. According to Lemma 40 and Lemma 39 there exist $p_b \in P_b^{(f)}|_x$ and $p_{\neg b} \in P_{\neg b}^{(f)}$ such that $q_1 = (p_{\neg b} + x)|_{0 \Leftarrow \mathtt{u}}$ and $q_0 = p_b + x$. Then, for every $i \in [n]$ the following holds:*

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

In words, for every $q_1 \in P_1^{(\mathrm{d}f(x;\cdot))}$ and $q_0 \in P_0^{(\mathrm{d}f(x;\cdot))}$, which label a row and a column of $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$, we can choose $p_b \in P_b^{(f)}|_x$ and $p_{\neg b} \in P_{\neg b}^{(f)}$, which label a row and a column of M_x, that yield q_1 and q_0. Moreover, the correspondence preserves the set of stable and different coordinates between (q_1, q_0) and $(p_b, p_{\neg b})$, hence the associated entries in $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$ and M_x are identical.

A concrete example is shown in Figures 2a and 2b, where the same colors as in Figure 3a are used for M_{1100} and $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f_{\mathrm{prime}}(1100;\cdot)}}$. It can be verified that $\mathtt{u}1\mathtt{u}0 \in P_0^{(f_{\mathrm{prime}})}$ yields $\mathtt{u}0\mathtt{u}0 \in P_0^{(\mathrm{d}f_{\mathrm{prime}}(1100;\cdot))}$ and $001\mathtt{u} \in P_1^{(f_{\mathrm{prime}})}$ yields $111\mathtt{u} \in P_1^{(\mathrm{d}f_{\mathrm{prime}}(1100;\cdot))}$ with the corresponding entries both equal to $\{2\}$.

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.

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 $(p_{\neg b} + x)|_{0 \Leftarrow \mathtt{u}} \in I_1^{(\mathrm{d}f(x;\cdot))} \setminus P_1^{(\mathrm{d}f(x;\cdot))}$ are super-rows. Implicants that are not prime, by definition, contain "extra" stable bits that can be replaced with $\mathtt{u}$ while preserving the output of the function. Therefore, these rows contain possibly more valid answers to the $\mathrm{KW}^{\mathtt{u}}$ on $\mathrm{d}f(x;\cdot)$ than the ones associated with $P_1^{(\mathrm{d}f(x;\cdot))}$. Figures 2a and 2b demonstrate this claim, as the super rows in 2a correspond to $\{\mathtt{u}011, 0\mathtt{u}11, 01\mathtt{u}1\}$, which yielded $\{\mathtt{u}111, 1\mathtt{u}11, 1\mathtt{u}\mathtt{u}1\}$ 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 $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$. Hence, M_x and $M_{\mathrm{KW}^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}$ 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 larger submatrix of $M_{\mathrm{KW}^{\mathtt{u}}}$.

▶ **Proposition 44.** *Let $f : \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$ such that $f(x) = b \in \{0,1\}$. We denote by M_x the submatrix of $M_{KW^{\mathtt{u}}_f}$ with labels $P_{\neg b}^{(f)} \times P_b^{(f)}|_x$. The following holds:*

$$\textit{mono-rec}(M_x) = \textit{mono-rec}(M_{KW^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}), \quad CC(M_x) = CC(M_{KW^{\mathtt{u}}_{\mathrm{d}f(x;\cdot)}}).$$

2.2 Proof Overview of Theorem 31

In this proof, we further analyze the communication matrix of the hazard-free KW game of f and show that if f is unate, then it can be reduced to the communication matrix of a suitable hazard-derivative. If f is not unate, we prove that any protocol must have more leaves than the optimal protocol for any of its hazard-derivatives.

In the first direction of Theorem 31, consider a unate f. We prove that there exists some $x' \in \{0,1\}^n$ such that $M_{x'} = M_{\mathrm{KW}_f^{\mathfrak{u}}}$, by characterizing the prime implicants and prime implicates of a unate function. Another way of proving this is to generalize [11, 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 [11, Lemma 4.4], that $f(x+c) = \mathrm{d}f(c;x)$.

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 $f(x) = b \in \{0,1\}$. In the case of f_{prime}, we examine the third most significant bit.[7] Let $s \in \{0,\mathfrak{u},1\}$, 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 $P_b^{d=x_d}, P_b^{d=\neg x_d}, P_{\neg b}^{d=x_d}, P_{\neg_b}^{d=\neg x_d} \neq \emptyset$ (x_i denote the i'th coordinate of x). We actually prove the following stronger claim:

▶ **Proposition 45.** *Let $f : \{0,1\}^n \to \{0,1\}$ 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.*

Indeed, in the case of f_{prime} and d as the third most significant bit, we have $p_0 := \mathfrak{u}\mathfrak{u}00$, $q_0 = 111\mathfrak{u} \in P_0^{(f_{\mathrm{prime}})}$ and $p_1 = 001\mathfrak{u}$, $q_1 = \mathfrak{u}101 \in P_1^{(f_{\mathrm{prime}})}$.

By Proposition 45, consider $p_{\neg b} \in P_b^{(f)}$ and $p_b \in P_b^{(f)}$ such that $(p_{\neg b})_d = x_d$ and $(p_b)_d = \neg x_d$. For every protocol solving $\mathrm{KW}_f^{\mathfrak{u}}$, there exists a monochromatic rectangle $\mathrm{Rec} \subseteq P_{\neg b}^{(f)} \times P_b^{(f)}$ in the partition induced by the protocol that contains $(p_{\neg b}, p_b)$, and so must be colored with d. In fact, $\mathrm{Rec} \subseteq P_{\neg b}^{d=x_d} \times P_b^{d=\neg x_d}$, since $(p_{\neg b})_d = x_d$ and $(p_b)_d = \neg x_d$. Hence, Rec is disjoint from M_x, as illustrated in Figure 3b. Removing the leaf associated with Rec from the protocol tree results in a protocol for M_x that has one less leaf, since $M_x \subseteq M_{\mathrm{KW}_f^{\mathfrak{u}}} \setminus \mathrm{Rec}$. This implies that there must be some gap in the number of leaves in the optimal protocols, thus proving the following proposition:

▶ **Proposition 46.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-unate Boolean function. Then for every $x \in \{0,1\}^n$, it holds that:*

$$\mathit{size}_F^+(\mathrm{d}f(x;\cdot)) < \mathit{size}_F^{\mathfrak{u}}(f).$$

2.3 Proof Overview of Theorem 34

We obtain two proofs for this Theorem that 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 two proofs are "dual" in some sense, so we focus on the second, which offers insights of independent interest. First, we prove the following lemma, stating that for a random Boolean function, with high probability, any hazard-derivative has a monotone formula that is not too large:

▶ **Lemma 47.** *A random Boolean function $f : \{0,1\}^n \to \{0,1\}$, sampled uniformly at random, satisfies that with high probability, for every $x \in \{0,1\}^n$:*

$$\mathit{size}_F^+(\mathrm{d}f(x;\cdot)) \leq \frac{n^{\log n}}{\log n^{(1-o(1))\log n}}.$$

[7] Note that $f(0000) = 0$, $f(0010) = 1$ and $f(1111) = 0$, $f(1101) = 1$.

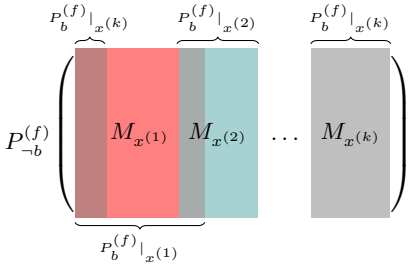


Figure 4 Decomposition of $M_{\mathrm{KW}_f^{\mathrm{u}}}$.

This stems from the observation that, with high probability, a random function is not constant on any large Boolean subcube, causing its hazard-derivatives to be mostly constant. Therefore, the prime implicants of its hazard-derivative tend to have a lot of unstable coordinates, allowing its DNF monotone formula to be fairly small.

The second step consists of proving Theorem 36, which gives a weak converse to Theorem 12. Building on the analysis of $M_{\mathrm{KW}_f^{\mathrm{u}}}$ discussed above, we prove that $M_{\mathrm{KW}_f^{\mathrm{u}}}$ 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 $p \overset{\mathrm{u}}{\leq} x$), we decompose $M_{\mathrm{KW}_f^{\mathrm{u}}}$ to k (potentially overlapping) M_x's with labels $P_{\neg b}^{(f)} \times P_b^{(f)}|_x$, as illustrated in Figure 4 (see Figure 2 for a concrete example). Although $M_{\mathrm{KW}^{\mathrm{u}}_{\mathrm{d}f(x;\cdot)}}$ may be a proper submatrix of M_x, Proposition 44 implies that each M_x is upper bounded by the complexity of $M_{\mathrm{KW}^{\mathrm{u}}_{\mathrm{d}f(x;\cdot)}}$.

By combining Theorem 36 with Lemma 47, we complete the proof of Theorem 34.

We note that the upper bound in Theorem 36 can be tight. The key observation is that if there exists $x^* \in \{0,1\}^n$ such that the prime implicates are covered by x^* and $\neg x^*$, then the upper bound is tight. 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.

This is the case for range functions: For $0 \leq a < b \leq n$ we call $R^n_{a,b} : \{0,1\}^n \to \{0,1\}$ a "range function", if $R^n_{a,b}(x) = 1$ if and only if $a \leq x_1 + \cdots + x_n < b$. We show that range functions have such x^*, and so, $R^n_{a,b}$-monochromatic rectangles are a subset of either M_{x^*} or $M_{\neg x^*}$. Therefore, there is an optimal protocol that begins by dividing $M_{\mathrm{KW}^{\mathrm{u}}_{R^n_{a,b}}}$ to M_{x^*} and $M_{\neg x^*}$.

2.4 Proof Overview of Propositions 37 and 38

For the first, we follow the ideas from the proof of the next lemma:

▶ **Lemma 48** ([14, Lemma 4]). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be monotone Boolean functions. Then:*

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

The proof of Lemma 48 uses two properties of $\mathrm{KW}^+_{f\diamond g}$: 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 $X, Y \in \{0, \mathrm{u}, 1\}^{n \times m}$, a valid answer (i, j) to $\mathrm{KW}^+_{f\diamond g}$ 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. 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.

Let $b \in \{0,1\}$, and let $P \in P_b^{(f \diamond g)}$. We think of P as a $n \times m$ ternary matrix. We denote by P_i the i'th row of P and by $\mathtt{u}^m$ a ternary input of length m which all of its coordinates are equal to $\mathtt{u}$. The following holds:

$$\exists p_b \in P_b^{(f)}, \quad (\tilde{g}(P_1), \ldots, \tilde{g}(P_n)) = p_b \quad \text{and} \quad \forall i \in [n], \quad P_i \in \begin{cases} P_1^{(g)}, & (p_b)_i = 1, \\ P_0^{(g)}, & (p_b)_i = 0, \\ \{\mathtt{u}^m\}, & \text{otherwise.} \end{cases}$$

Given the above, if $P_1^{(f)} \subseteq \{1, \mathtt{u}\}^n$ and $P_0^{(f)} \subseteq \{0, \mathtt{u}\}^n$, a valid answer to the hazard-free KW game of $f \diamond g$ on the inputs $X, Y \in \{0, \mathtt{u}, 1\}^{nm}$ 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 can apply the same idea as [14, Lemma 4] to obtain:

▶ **Lemma 49** (Direct sum is reducible to $\mathsf{KW}^{\mathtt{u}}_{f \diamond g}$). *Let $f : \{0,1\}^n \to \{0,1\}$ be a* monotone *Boolean function, and let $g : \{0,1\}^m \to \{0,1\}$ be a Boolean function. Then, it holds that:*

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

The proof of Proposition 38 follows by substituting known upper bounds into Proposition 37.

3 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 50.** *Is it true that for every Boolean function $f : \{0,1\}^n \to \{0,1\}$, $size_F^{\mathtt{u}}(f) \leq 2^{(1+o(1))n}$?*

Another interesting question is whether Theorem 31 also holds for circuits:

▶ **Question 51.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant Boolean function. Is it true that there exists $x \in \{0,1\}^n$ such that:*

$$size_C^+(\mathrm{d}f(x;\cdot)) = size_C^{\mathtt{u}}(f),$$

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 [14, 9, 5, 8, 18, 21], 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 28** (Hazard-Free KRW conjecture). *Let $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$ be non-constant Boolean functions. Then the following holds:*

$$depth_F^{\mathtt{u}}(f \diamond g) \approx depth_F^{\mathtt{u}}(f) + depth_F^{\mathtt{u}}(g).$$

Finally, we consider block compositions with the XOR function. Recall that the hazard-free complexity of XOR_n is equal to its standard formula complexity. Additionally, its prime implicants and prime implicates 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 size_F with $\mathrm{size}_F^{\mathrm{u}}$.

▶ **Theorem 52** (Composition over parity [4, Theorem 3.1]). *Let $f : \{0,1\}^k \to \{0,1\}$ be a non-constant Boolean function. Then:*

$$size_F(f \diamond XOR_m) \geq \frac{size_F(f) \cdot size_F(XOR_m)}{2^{\tilde{O}(\sqrt{k+\log m})}},$$

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

References

1. Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7:1–22, 1987. doi:10.1007/BF02579196.
2. Leah London Arazi and Amir Shpilka. On the complexity of hazard-free formulas, 2025. arXiv:2411.09026.
3. Johannes Bund, Christoph Lenzen, and Moti Medina. Optimal metastability-containing sorting via parallel prefix computation. *IEEE Transactions on Computers*, 69(2):198–211, 2020. doi:10.1109/TC.2019.2939818.
4. Irit Dinur and Or Meir. Toward the krw composition conjecture: Cubic formula lower bounds via communication complexity. *Computational Complexity*, 27:375–462, September 2018. doi:10.1007/s00037-017-0159-x.
5. Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiří Sgall. Communication complexity towards lower bounds on circuit depth. *Comput. Complexity*, 10(3):210–246, 2001. doi:10.1007/s00037-001-8195-x.
6. Edward B. Eichelberger. Hazard detection in combinational and sequential switching circuits. *IBM Journal of Research and Development*, 9(2):90–99, 1965. doi:10.1147/rd.92.0090.
7. Stephan Friedrichs, Matthias Függer, and Christoph Lenzen. Metastability-containing circuits. *IEEE Transactions on Computers*, 67(8):1167–1183, 2018. doi:10.1109/TC.2018.2808185.
8. Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: The composition of a function and a universal relation. *SIAM Journal on Computing*, 46(1):114–131, 2017. doi:10.1137/15M1018319.
9. Johan Håstad and Avi Wigderson. Composition of the universal relation. In *Advances in computational complexity theory (New Brunswick, NJ, 1990)*, volume 13 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 119–134. Amer. Math. Soc., Providence, RI, 1993. doi:10.1090/dimacs/013/07.
10. David A. Huffman. The design and use of hazard-free switching networks. *J. ACM*, 4(1):47–62, January 1957. doi:10.1145/320856.320866.
11. Christian Ikenmeyer, Balagopal Komarath, Christoph Lenzen, Vladimir Lysikov, Andrey Mokhov, and Karteek Sreenivasaiah. On the complexity of hazard-free circuits. *J. ACM*, 66(4), August 2019. doi:10.1145/3320123.
12. Christian Ikenmeyer, Balagopal Komarath, and Nitin Saurabh. Karchmer-Wigderson Games for Hazard-Free Computation. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.74.
13. Stasys Jukna. Notes on hazard-free circuits. *SIAM Journal on Discrete Mathematics*, 35(2):770–787, 2021. doi:10.1137/20M1355240.

14 Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5:191–204, 1995. doi:10.1007/BF01206317.

15 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990. doi:10.1137/0403021.

16 Stephen Cole Kleene. *Introduction to metamathematics*. D. Van Nostrand Co., Inc., New York, 1952.

17 Balagopal Komarath and Nitin Saurabh. On the complexity of detecting hazards. *Inf. Process. Lett.*, 162:105980, 2020. doi:10.1016/J.IPL.2020.105980.

18 Sajin Koroth and Or Meir. Improved composition theorems for functions and relations. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 48:1–48:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.APPROX-RANDOM.2018.48.

19 Edward J. McCluskey. *Logic design principles - with emphasis on testable semicustom circuits*. Prentice Hall series in computer engineering. Prentice Hall, 1986.

20 EJ McCluskey. Transients in combinational logic circuits. *Redundancy Technique for Computing System*, 9, 1962.

21 Or Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Computational Complexity*, 29(1):4, 2020. doi:10.1007/s00037-020-00194-8.

22 Masao Mukaidono. On the B-ternary logical function—A ternary logic considering ambiguity. *Syst. Comput. Controls*, 3(3):27–36, 1972.

23 David E. Muller. Complexity in electronic switching circuits. *IRE Transactions on Electronic Computers*, EC-5(1):15–19, 1956. doi:10.1109/TEC.1956.5219786.

24 Steven M. Nowick and David L. Dill. 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):986–997, 1995. doi:10.1109/43.402498.

25 Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean function. *Soviet Math. Dokl.*, 31:354–357, 1985.

26 Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. Krw composition theorems via lifting. *Computational Complexity*, 33(1):1–97, 2024. doi:10.1007/s00037-024-00250-7.

27 John Riordan and Claude E. Shannon. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942. doi:10.1002/sapm194221183.

28 Claude E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

29 Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8:141–142, 1988. doi:10.1007/BF02122563.

30 S. Unger. Hazards and delays in asynchronous sequential switching circuits. *IRE Transactions on Circuit Theory*, 6(1):12–25, 1959. doi:10.1109/TCT.1959.1086527.

31 Michael Yoeli and Shlomo Rinon. Application of ternary algebra to the study of static hazards. *J. ACM*, 11(1):84–97, January 1964. doi:10.1145/321203.321214.