Boolean Circuit Complexity and Two-Dimensional Cover Problems

Bruno Cavalar[*]
Department of Computer Science
University of Warwick

Igor C. Oliveira[†]
Department of Computer Science
University of Warwick

July 31, 2024

Abstract

We reduce the problem of proving deterministic and nondeterministic Boolean circuit size lower bounds to the analysis of certain two-dimensional combinatorial cover problems. This is obtained by combining results of Razborov (1989), Karchmer (1993), and Wigderson (1993) in the context of the fusion method for circuit lower bounds with the graph complexity framework of Pudlák, Rödl, and Savický (1988). For convenience, we formalize these ideas in the more general setting of "discrete complexity", i.e., the natural set-theoretic formulation of circuit complexity, variants of communication complexity, graph complexity, and other measures.

We show that random graphs have linear graph cover complexity, and that explicit super-logarithmic graph cover complexity lower bounds would have significant consequences in circuit complexity. We then use discrete complexity, the fusion method, and a result of Karchmer and Wigderson (1993) to introduce nondeterministic graph complexity. This allows us to establish a connection between graph complexity and nondeterministic circuit complexity.

Finally, complementing these results, we describe an exact characterization of the power of the fusion method in discrete complexity. This is obtained via an adaptation of a result of Nakayama and Maruoka (1995) that connects the fusion method to the complexity of "cyclic" Boolean circuits, which generalize the computation of a circuit by allowing cycles in its specification.

[*]E-mail: brunocavalar@gmail.com
[†]E-mail: igor.oliveira@warwick.ac.uk

Contents

1 Introduction

1.1 Overview

Obtaining circuit size lower bounds for explicit Boolean functions is a central research problem in theoretical computer science. While restricted classes of circuits such as constant-depth circuits and monotone circuits are reasonably well understood (see, e.g., [Juk12]), understanding the power and limitations of general (unrestricted) Boolean circuits remains a major challenge.

The strongest known lower bounds on the number of gates necessary to compute an explicit Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ are of the form $C \cdot n$ for a constant $C \leq 5$. The largest known value of C depends on the exact set of allowed operations (see [LY22, FGHK16] and references therein). To the best of our knowledge, the existing lower bounds on gate complexity for unrestricted Boolean circuits with a single output bit have all been obtained via the gate elimination method and its extensions. Unfortunately, it is not expected that this technique can lead to much better bounds [GHKK16], let alone super-linear circuit size lower bounds.

This paper revisits a classical approach to lower bounds known as the fusion method [Raz89, Kar93]. The latter reduces the analysis of the circuit complexity of a Boolean function to obtaining bounds on certain related combinatorial cover problems. The method can also be adapted to weaker circuit classes, where it has been successful in some contexts (see [Wig93] for an overview of results).[1]

An advantage of the fusion method over the gate elimination method is that it provides a tight characterization (up to a constant or polynomial factor, depending on the formulation) of the circuit complexity of

[1] The fusion method can be seen as an instantiation of the generalized approximation method. For a self-contained exposition of the connection between the fusion method and the approximation method, we refer the reader to [Oli18].

a function. In particular, if a strong enough circuit lower bound holds, then in principle it can be established via the fusion method.

Contributions. We can informally summarize our contributions as follows:

1. We exhibit a new instantiation of the fusion method that reduces the problem of proving deterministic and nondeterministic Boolean circuit size lower bounds to the analysis of "two-dimensional" combinatorial cover problems.

2. To achieve this, we introduce a framework that combines the fusion method for lower bounds with the notion of graph complexity and its variants [PRS88, Juk13]. In particular, we observe that cover complexity offers a particularly strong "transference" theorem between Boolean circuit complexity and graph complexity.

3. As a byproduct of our conceptual and technical contributions, we obtain a tight asymptotic bound on the cover complexity of a random graph, and introduce a useful notion of nondeterministic graph complexity.

4. Finally, we describe an exact correspondence between cover complexity and circuit complexity. This is relevant for the investigation of state-of-the-art circuit lower bounds of the form $C \cdot n$, where C is constant.

In the next section, we describe these results and their connections to previous work in more detail.

1.2 Results

Notation. Given a family $\mathcal{B} = \{B_1, \ldots, B_m\}$, where each set B_i is contained in a finite fixed ground set Γ, and a target set A, we let $D(A \mid \mathcal{B})$ denote the minimum total number of pairwise unions and intersections needed to construct A starting from $B_1, \ldots, B_m$. We say that $D(A \mid \mathcal{B})$ is the *discrete complexity* of A with respect to $\mathcal{B}$ (see Section 2.1 for a formal presentation). We will be interested in the discrete complexity of *non-trivial* sets A, i.e., when $A \neq \emptyset$ and $A \neq \Gamma$.

This general definition can be used to capture a variety of problems. For instance, the monotone circuit complexity of a function $f \colon \{0,1\}^n \to \{0,1\}$ is simply $D(f^{-1}(1) \mid \{x_1, \ldots, x_n, \emptyset, \bar{1}\})$, where each symbol from $\{x_1, \ldots, x_n, \emptyset, \bar{1}\}$ represents the natural corresponding subset of $\{0,1\}^n$. Similarly, we can capture (non-monotone) Boolean circuit complexity by considering the family $\mathcal{B}_n = \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$ of subsets of $\{0,1\}^n$ and the corresponding complexity measure $D(f^{-1}(1) \mid \mathcal{B}_n)$.

Let $N = 2^n$ for some $n \in \mathbb{N}$, and let $[N] = \{1, 2, \ldots, N\}$. As another example in discrete complexity, we can consider subsets $R_1, \ldots, R_N, C_1, \ldots, C_N$ of the ground set $[N] \times [N]$, where each set $R_i = \{(i,j) \mid j \in [N]\}$ corresponds to the i-th "row", and each set $C_j = \{(i,j) \mid i \in [N]\}$ corresponds to the j-th "column". Then, given a set $G \subseteq [N] \times [N]$ and $\mathcal{G}_{N,N} = \{R_1, \ldots, R_N, C_1, \ldots, C_N\}$, the quantity $D(G \mid \mathcal{G}_{N,N})$ is known as the *graph complexity* of G (see [PRS88, Juk13]).

For the discussion below, we will need another definition. We let $D_\cap(A \mid \mathcal{B})$ denote the minimum number of pairwise intersections sufficient to construct A from the sets in $\mathcal{B}$. We say that $D_\cap(A \mid \mathcal{B})$ is the *intersection complexity* of A with respect to $\mathcal{B}$. We refer to Figure 1 for an example. It is possible to show that $D_\cap(A \mid \mathcal{B})$ and $D(A \mid \mathcal{B})$ are polynomially related, with a dependency on $|\mathcal{B}|$ (see Section 2.3 for more details).

Given an arbitrary set A and a family $\mathcal{B}$ as above, one can introduce a complexity measure $\rho(A, \mathcal{B})$ that is closely related to $D(A \mid \mathcal{B})$. In more detail, we define an appropriate bipartite graph $\Phi_{A,\mathcal{B}} = (V_{\mathsf{pairs}}, V_{\mathsf{filters}}, \mathcal{E})$, called the *cover graph* of A and $\mathcal{B}$, and let $\rho(A, \mathcal{B})$ denote the minimum number of vertices in V_{pairs} whose adjacent edges cover all the vertices in V_{filters}. (Since the definition of the graph $\Phi_{A,\mathcal{B}}$

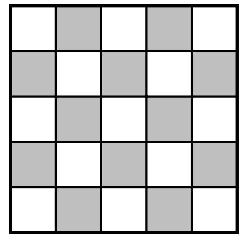

Figure 1: A graphical representation of a set $G \subseteq [5] \times [5]$ of intersection complexity $D_\cap(G \mid \mathcal{G}_{5,5}) \leq 2$ via $G = \big((R_2 \cup R_4) \cap (C_1 \cup C_3 \cup C_5)\big) \cup \big((C_2 \cup C_4) \cap (R_1 \cup R_3 \cup R_5)\big)$.

is somewhat technical and won't be needed in the subsequent discussion, it is deferred to Section 3.1). We say that $\rho(A, \mathcal{B})$ is the *cover complexity* of A with respect to $\mathcal{B}$.

Our first observation is that, by a straightforward adaptation of the fusion method for lower bounds [Raz89, Kar93, Wig93] to our framework, the following relation holds:

$$\rho(A, \mathcal{B}) \ \leq \ D_\cap(A \mid \mathcal{B}) \ \leq \ \rho(A, \mathcal{B})^2. \tag{1}$$

In particular, cover complexity provides a lower bound on intersection complexity. We are particularly interested in applications of the inequalities above to graph complexity. There are two main reasons for this. Firstly, to each graph $G \subseteq [N] \times [N]$ one can associate a natural Boolean function $f_G \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ (see Section 2.4), where $N = 2^n$, and it is known that lower bounds on the graph complexity of G yield lower bounds on the Boolean circuit complexity of f_G [PRS88]. (There can be a significant loss on the parameters of such transference results depending on the context. We refer to [Juk13] for more details. See also the discussion before Remark 14 below.) Secondly, the cover problem defining $\rho(G, \mathcal{G}_{N,N})$ involves a two-dimensional ground set $[N] \times [N]$, in contrast to the n-dimensional ground set $\{0,1\}^n$ found in Boolean function complexity. We hope this perspective can inspire new techniques, and indeed we show how this perspective can be used to give a tight bound for a natural Boolean function in Section 4.2.

Our second observation is that a tight connection can be established between graph complexity and Boolean circuit complexity by focusing on intersection complexity and cover complexity.

Lemma 1 (Transference of Lower Bounds)**.** *For every non-trivial bipartite graph $G \subseteq [N] \times [N]$ and corresponding Boolean function $f_G \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, we have*

$$\rho(f_G^{-1}(1), \mathcal{B}_{2n}) \ \geq \ \rho(G, \mathcal{G}_{N,N}), \ \textit{and} \tag{2}$$
$$D(f_G^{-1}(1) \mid \mathcal{B}_{2n}) \ \geq \ D_\cap(G \mid \mathcal{G}_{N,N}). \tag{3}$$

The second inequality is implicit in the literature on graph complexity. We include it in the statement of Lemma 1 for completeness. Using Lemma 1, Equation (1), and another idea, we note in Section 2.4 that a lower bound of the form $C \cdot \log N$ on $\rho(G, \mathcal{G}_{N,N})$ yields a lower bound of the form $C \cdot m - O(1)$ on the AND complexity of a related function $F \colon \{0,1\}^m \to \{0,1\}$. It is worth noting that lower bounds of the form Cn for $C > 1$ on the AND complexity of explicit Boolean functions can be obtained using gate-elimination techniques [Gol18], so the problem considered here does not suffer from a "barrier" at n gates as in the setting of multiplicative complexity [Sch88]. We leave open the problem of matching (or more ambitiously strengthening) existing Boolean circuit lower bounds obtained via gate elimination using our framework.

Complementing the approach to non-trivial circuit lower bounds discussed above, we show the following result for non-explicit graphs.

Theorem 2 (Cover complexity of a random graph)**.** *Let $N = 2^n$, and let $G \subseteq [N] \times [N]$ be a uniformly random bipartite graph. Then, asymptotically almost surely,*

$$\rho(G, \mathcal{G}_{N,N}) \;=\; \Theta(N).$$

Since the state of the art in Boolean circuit lower bounds is of the form $C \cdot n$ for a small constant C, the discussion above motivates the investigation of a tighter version of Equation (1). Next, we show that cover complexity can be *exactly* characterized using the complexity of *cyclic constructions*. Roughly speaking, $D^{\circlearrowright}(A \mid \mathcal{B})$ denotes the minimum number of unions and intersections in a cyclic construction of A from sets in $\mathcal{B}$, where a cyclic construction can be seen as the analogue of a Boolean circuit allowed to contain cycles. We refer to Section 2.5 for the definition. Similarly, we can also consider $D^{\circlearrowright}_{\cap}(A \mid \mathcal{B})$, the intersection complexity of cyclic constructions.

Theorem 3 (Exact characterization of cover complexity)**.** *Let $A \subseteq \Gamma$ be a non-trivial set, and let $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$ be a non-empty family of sets. Then*

$$\rho(A, \mathcal{B}) \;=\; D^{\circlearrowright}_{\cap}(A \mid \mathcal{B}).$$

This precise correspondence is obtained by refining an idea from [NM95], which obtained a characterization of a variant of cover complexity up to a constant factor. There are some technical differences though. In contrast to their work, here we consider (monotone) semi-filters instead of a more general class of functionals $\mathcal{F} \subseteq \mathcal{P}(U)$ in the definition of cover complexity, and intersection complexity instead of Boolean circuit complexity. Additionally, the result is presented in the set-theoretic framework of the fusion method (which is closer to our notion of discrete complexity), while [NM95] employed a formulation via legitimate models and the generalized approximation method.

As an immediate consequence of Theorem 3 and a cover complexity lower bound from [Kar93], it follows that every monotone *cyclic* Boolean circuit that decides if an input graph on n vertices contains a triangle contains at least $\Omega(n^3/(\log n)^4)$ fan-in two AND gates.[2] We refer to Section 3.4 for more details.

The tight bound in Theorem 3 highlights a mathematical advantage of the investigation of cyclic constructions and cyclic Boolean circuits. Interestingly, the strongest known lower bounds against unrestricted (non-monotone) Boolean circuits obtained via the gate elimination method [LY22, FGHK16] also incorporate concepts related to cyclic computations.

Our last contribution is of a conceptual nature. The fusion method offers a different yet equivalent formulation of circuit complexity. This allows us to port some of the abstractions and characterizations provided by different notions of cover complexity to the setting of discrete complexity. As an example, we introduce *nondeterministic graph complexity* through a dual notion of "nondeterministic" cover complexity from [Kar93], and show a simple application to nondeterministic Boolean circuit lower bounds via a transference lemma for nondeterministic complexity.[3]

Going beyond the contrast between state-of-the-art lower bounds for monotone and non-monotone computations, it would also be interesting to obtain an improved understanding of which settings of discrete complexity are susceptible to strong unconditional lower bounds.

Organization. The main definitions are given in Section 2. To make the paper self-contained, we include a proof of Equation (1) in Section 3. The proof of Lemma 1 appears in Section 2.4 and Section 4.1. The proof of Theorem 2 is presented in Section 4.1, while the proof of Theorem 3 is given in Section 3.4.

[2] This consequence does not immediately follow from the work of [NM95], as their formulation is not consistent with the use of monotone functionals employed in the definition of ρ followed here and in [Kar93].

[3] Observe that the definition of nondeterministic complexity for Boolean functions relies on Boolean circuits extended with extra input variables. It is not obvious how to introduce a natural analogue in the context of graph complexity, which relies on graph constructions.

Finally, a discussion on nondeterministic graph complexity and a simple application of this notion appear in Section 4.3.

Acknowledgements. We would like to thank Sasha Golovnev and Rahul Santhanam for discussions about the AND complexity of Boolean functions. This work received support from the Royal Society University Research Fellowship URF\R1\191059; the UKRI Frontier Research Guarantee Grant EP/Y007999/1; and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick.

2 Discrete Complexity

2.1 Definitions and notation

We adopt the convention that $\mathbb{N} \stackrel{\text{def}}{=} \{0, 1, 2, \ldots\}$, $\mathbb{N}^+ \stackrel{\text{def}}{=} \mathbb{N} \setminus \{0\}$, $[t] \stackrel{\text{def}}{=} \{1, \ldots, t\}$, where $t \in \mathbb{N}^+$, and $\mathcal{P}(\cdot)$ is the power-set construction.

Let Γ be a nonempty finite set. We refer to this set as the *ground set*, or the *ambient space*. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$ be a family of subsets of Γ. We say that a set $B_i \in \mathcal{B}$ is a *generator*. Given a set $A \subseteq \Gamma$, we are interested in the minimum number of elementary set operations necessary to construct A from the generator sets in $\mathcal{B}$. The allowed operations are *union* and *intersection*. Formally, we let $D(A \mid \mathcal{B})$ be the minimum number $t \geq 1$ such that there exists a *sequence* $A_1, \ldots, A_t$ of sets contained in Γ for which the following holds: $A_t = A$, and for every $i \in [t]$, A_i is either the union or the intersection of two (not necessarily distinct) sets in $\mathcal{B} \cup \{A_1, \ldots, A_{i-1}\}$. We say that a sequence of this form *generates* A from $\mathcal{B}$. If there is no finite t for which such a sequence exists, then $D(A \mid \mathcal{B}) \stackrel{\text{def}}{=} \infty$.[4] Consequently, $D \colon \mathcal{P}(\Gamma) \times \mathcal{P}(\mathcal{P}(\Gamma)) \to \mathbb{N}^+ \cup \{\infty\}$. We say that $D(A \mid \mathcal{B})$ is the *discrete complexity* of A with respect to $\mathcal{B}$.

We use $D_\cap(A \mid \mathcal{B})$ to denote the minimum number of *intersections* in any sequence that generates A from $\mathcal{B}$. The value $D_\cup(A \mid \mathcal{B})$ is defined analogously. We will often refer to these measures as *intersection complexity* and *union complexity*, respectively.

Fact 4. *If $A \in \mathcal{B}$, then $D(A \mid \mathcal{B}) = 1$ and $D_\cap(A \mid \mathcal{B}) = D_\cup(A \mid \mathcal{B}) = 0$.*

We have the following obvious inequality, which in general does not need to be tight (Fact 4 offers a trivial example).

Fact 5. $D(A \mid \mathcal{B}) \geq D_\cap(A \mid \mathcal{B}) + D_\cup(A \mid \mathcal{B})$.

When the ambient space Γ is clear from the context, we let $E^c \subseteq \Gamma$ denote the complement of a set $E \subseteq \Gamma$. For convenience, for a set $U \subseteq \Gamma$, we use B_U as a shorthand for $B \cap U$. For a family of sets $\mathcal{B}$, we let $\mathcal{B}_U \stackrel{\text{def}}{=} \{B_U \mid B \in \mathcal{B}\}$.

Let $A_1, \ldots, A_t$ be a sequence of sets that generates A from $\mathcal{B}$, where $|\mathcal{B}| = m$. It will be convenient in some inductive proofs to consider the *extended sequence* $B_1, \ldots, B_m, A_1, \ldots, A_t$ that includes as a prefix the generators from $\mathcal{B}$. The particular order of the sets B_i is not relevant. While the extended sequence has length $m + t$, we will refer to it as a sequence of complexity t. Similarly, if the number of intersections employed in the definition of the sequence is k, we say it has intersection complexity k.

Given a construction of A from $\mathcal{B}$ specified by a sequence $A_1, \ldots, A_t$ and its corresponding union and intersection operations, we let Λ be the *set of intersections* in the sequence, where we represent an intersection operation $A_\ell = A_i \cap A_j$ by the pair (A_i, A_j).

[4]A simple example is that of a non-monotone Boolean function represented by $A \subseteq \{0,1\}^n$ and $\mathcal{B}$ as the family of generators in monotone circuit complexity.

For an ambient space Γ and $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$, we use $\langle \Gamma, \mathcal{B} \rangle$ to represent the corresponding *discrete space*. We assume for simplicity that $\Gamma = \bigcup_{B \in \mathcal{B}} B$. We extend the notation introduced above, and use $D(A_1, \ldots, A_\ell \mid \mathcal{B})$ to denote the discrete complexity of simultaneously generating $A_1, \ldots, A_\ell$ from $\mathcal{B}$. In other words, this is the minimum number t such that there exists a sequence $E_1, \ldots, E_t$ of sets contained in Γ such that every set A_i appears in the sequence at least once, and each E_j is obtained from the preceding sets in the sequence and the sets in $\mathcal{B}$ either by a union or by an intersection operation.

Finally, note that we tacitly assume in most proofs presented in this section that $D(A \mid \mathcal{B})$ is finite, as otherwise the corresponding statements are trivially true. We will also assume in these statements that $A \subseteq \bigcup_{B \in \mathcal{B}} B = \Gamma$ in order to avoid trivial considerations.

2.2 Examples

2.2.1 Boolean circuit complexity

This is the classical setting where for each $n \in \mathbb{N}^+$, $\Gamma = \{0,1\}^n$ is the set of vertices of the n-dimensional hypercube, A corresponds to $f^{-1}(1)$ for a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, and $\mathcal{B} = \{B_1, \ldots, B_n, B_1^c, \ldots, B_n^c\}$, where $B_i = \{v \in \Gamma \mid v_i = 1\}$. By definition, $D(A \mid \mathcal{B})$ captures the *circuit complexity* of f. If we drop the generators B_i^c from the family $\mathcal{B}$, and add the sets $\emptyset$ and $\bar{1} \stackrel{\text{def}}{=} \{0,1\}^n$ to it, we get *monotone circuit complexity* instead of circuit complexity.

2.2.2 Bipartite graph complexity

Let $\Gamma = [N] \times [M]$, where $N, M \in \mathbb{N}^+$. A set $G \subseteq \Gamma$ can be viewed either as a bipartite graph with parts $L = [N]$ and $R = [M]$, or as an $N \times M$ $\{0,1\}$-valued matrix. We let $R_i \subseteq [N] \times [M]$ denote the matrix with 1's in the i-th row, and 0's elsewhere. Similarly, $C_j \subseteq [N] \times [M]$ denotes the matrix with 1's in the j-th column, and 0's elsewhere. (Each R_i and C_j is called a *star* in graph terminology). We let $\mathcal{G}_{N,M} = \{R_1, \ldots, R_N, C_1, \ldots, C_M\}$. The value $D(G \mid \mathcal{G}_{N,M})$ is known as the *star complexity* of G ([PRS88], see also [Juk13] and references therein). We will refer to it simply as *graph complexity*. Notice that, for every non-empty graph G, $D_\cap(G \mid \mathcal{G}_{N,M}) \leq \min\{N, M\}$.

We remark that a related notion of *clique complexity* is discussed in [Juk12]. In this notion, the generators are sets of the form $W_S := \bigcup_{i \in S} R_i$ and $Z_T := \bigcup_{j \in T} C_j$, for some $S \subseteq [N]$ and $T \subseteq [M]$. Let $\mathcal{K}_{N,M} = \{W_S : S \subseteq [N]\} \cup \{Z_T : T \subseteq [M]\}$. Note that the intersection clique complexity of a graph G is *equal* to its intersection graph complexity (i.e., $D_\cap(G \mid \mathcal{K}_{N,M}) = D_\cap(G \mid \mathcal{G}_{N,M})$).[5]

One can also consider the graph complexity of *non-bipartite* graphs via an appropriate choice of generators (as in, e.g., [Juk13]), though we will not be concerned with this variant in this work.

2.2.3 Higher-dimensional generalizations of graph complexity

This is the natural extension of the ambient space $[N] \times [N]$ to $[N]^d$, where $d \in \mathbb{N}^+$ is a fixed dimension. Every generator contained in $[N]^d$ is a set of elements described by a sequence of the form $(\star, \ldots, \star, a, \star, \ldots, \star)$, where an element $a \in [N]$ is fixed in exactly one coordinate. We let $\mathcal{G}_N^{(d)}$ be the corresponding family of generators. Notice that $|\mathcal{G}_N^{(d)}| = dN$. Given a d-dimensional tensor $A \subseteq [N]^d$, we denote its *d-dimensional graph complexity* by $D(A \mid \mathcal{G}_N^{(d)})$.

To some extent, graph complexity and Boolean circuit complexity are extremal examples of non-trivial discrete spaces, in the sense that the former minimizes the number of dimensions and maximizes the possible

[5] We also remark that the *decision tree clique complexity* of a graph G (in which we are allowed to query an arbitrary generator from $\mathcal{K}_{N,M}$) is known to capture *exactly* the communication complexity of an associated function f_G [PRS88, Section 3].

values in each coordinate, while the latter does the opposite. The higher dimensional graphs generalize both cases.

2.2.4 Combinatorial rectangles from communication complexity

The domain is $[N] \times [N]$, and its associated family $\mathcal{R}_{N,N}$ of generators contains every *combinatorial rectangle* $R = U \times V$, where $U, V \subseteq [N]$ are arbitrary subsets. In particular, $|\mathcal{R}_{N,N}| = 2^{2N}$, while the number of subsets of $[N] \times [N]$ is 2^{N^2}. Observe that $\mathcal{R}_{N,N}$ extends the set of generators employed in graph complexity. Consequently, for $G \subseteq [N] \times [N]$, $D(G \mid \mathcal{R}_{N,N}) \leq D(G \mid \mathcal{G}_{N,N})$. Moreover, $D_\cap(G \mid \mathcal{R}_{N,N}) = 0$ for every graph.

Observe that there is an interesting contrast among all these different spaces: the ratio between the *size of the ambient space* and *the number of generators*. For instance, in graph complexity the two are polynomially related, in Boolean circuits the ambient space is exponentially larger, and in the discrete space involving combinatorial rectangles the opposite happens. These natural discrete spaces exhibit three important regimes of parameters in discrete complexity.

2.3 Basic lemmas and other useful results

By combining sequences, we have the following trivial inequality.

Fact 6. *For every set $E \subseteq \Gamma$ and $\diamond \in \{\cap, \cup\}$, $D_\diamond(A \mid \mathcal{B}) \leq D_\diamond(A \mid E, \mathcal{B}) + D_\diamond(E \mid \mathcal{B})$.*[6]

Proof. Let $t_1 = D_\diamond(A \mid E, \mathcal{B})$, witnessed by the sequence $A_1, \ldots, A_{t_1}$. Also, let $t_2 = D_\diamond(E \mid \mathcal{B})$, with a corresponding sequence $E_1, \ldots, E_{t_2}$. Then $E_1, \ldots, E_{t_2}, A_1, \ldots, A_{t_1}$ is a sequence of length $t_1 + t_2$ showing that $D_\diamond(A \mid \mathcal{B}) \leq t_1 + t_2$. □

Observe that a construction of an arbitrary set A from $\mathcal{B}$ provides a construction of A_U from the sets in $\mathcal{B}_U$ (recall that $A_U \stackrel{\text{def}}{=} A \cap U$, etc.). Indeed, it is easy to see that if $A^1, \ldots, A^t$ generates A from $\mathcal{B}$, then $A^1_U, \ldots, A^t_U$ generates A_U from $\mathcal{B}_U$.

Fact 7. $D(A_U \mid \mathcal{B}_U) \leq D(A \mid \mathcal{B})$.

For convenience, we say that $A^1_U, \ldots, A^t_U$ is the *relativization* of the sequence $A^1, \ldots, A^t$ with respect to U.

The following simple technical fact will be useful. The proof is an easy induction via extended sequences.

Fact 8. *If A and $\mathcal{B}$ are non-empty, then $D_\cap(A \mid \mathcal{B}) = D_\cap(A \mid \mathcal{B} \cup \{\emptyset\})$.*

The next lemma shows that intersection complexity and discrete complexity are polynomally related, with a dependency on $|\mathcal{B}|$. This was first observed for monotone circuits in [AB87].

Lemma 9 (Immediate from [Zwi96])**.** *If $1 < D_\cap(A \mid \mathcal{B}) = k < \infty$, then*

$$D(A \mid \mathcal{B}) = O(k(|\mathcal{B}| + k)/\log k).$$

We describe a self-contained, indirect proof of a weaker form of this lemma in Section 3.3 (Corollary 28).

[6] We often abuse notation and write $D(A \mid E, \mathcal{B})$ instead of $D(A \mid \{E\} \cup \mathcal{B})$.

Given A and $\mathcal{B}$, there is a simple test to decide if $D(A \mid \mathcal{B})$ is finite, i.e., if there exists a finite sequence that generates A from $\mathcal{B}$. Let $\mathcal{B} = \{B_1, \ldots, B_m\}$. Given $w \in \Gamma$, we let $\mathsf{vec}(w) \in \{0,1\}^m$ be the vector with $\mathsf{vec}(w)_i = 1$ if and only if $w \in B_i$. For a set $C \subseteq \Gamma$, let $\mathsf{vec}(C) = \{\mathsf{vec}(c) \mid c \in C\}$. For vectors $u, v \in \{0,1\}^n$, we write $u \preceq v$ if $u_i \leq v_i$ for each $i \in [m]$.[7]

Proposition 10 (Finiteness test)**.** *$D(A \mid \mathcal{B})$ is finite if and only if there are no vectors $u \in \mathsf{vec}(A)$ and $v \in \mathsf{vec}(A^c)$ such that $u \preceq v$.*

Proof. Let $a \in A$ and $b \in A^c$ be elements such that $u = \mathsf{vec}(a) \preceq \mathsf{vec}(b) = v$. Suppose there is a construction $A_1, \ldots, A_t$ of A from $\mathcal{B}$. It follows easily by induction that $b \in A_t$, which is contradictory. On the other hand, if there is no element b and vector v with this property, it is not hard to see that $A = \bigcup_{u \in \mathsf{vec}(A)} \bigcap_{i : u_i = 1} B_i$. This completes the proof of the proposition. □

Finally, observe that standard counting arguments yield the existence of sets of high discrete complexity.

Lemma 11 (Complex sets)**.** *Let $k = |\Gamma|$ and $m = |\mathcal{B}|$. If $3s\lceil \log(m+s) \rceil < k$, there exists a set $A \subseteq \Gamma$ such that $D(A \mid \mathcal{B}) \geq s$.*

For instance, a random matrix $M \subseteq [N] \times [N]$ satisfies $D(M \mid \mathcal{R}_{N,N}) = \Omega(N)$, while a random graph $G \subseteq [N] \times [N]$ has $D(G \mid \mathcal{G}_{N,N}) = \Omega(N^2/\log N)$. It is easy to see that the former lower bound is asymptotically tight. The tightness of the graph complexity bound is also known (cf. [Juk13, Theorem 1.7]).

2.4 Transference of lower bounds

The following lemma generalizes a similar reduction from graph complexity (see, e.g., [Juk13, Section 1.3]).

Lemma 12. *Let $\langle \Gamma_1, \mathcal{B}_1 \rangle$ and $\langle \Gamma_2, \mathcal{B}_2 \rangle$ be discrete spaces, and $\phi \colon \Gamma_1 \to \Gamma_2$ be an injective function. Assume that $\mathcal{B}_2 = \{B_1^2, \ldots, B_m^2\}$. Then, for every $A_1 \subseteq \Gamma_1$,*

$$\begin{aligned} D(\phi(A_1) \mid \mathcal{B}_2) &\geq D(A_1 \mid \mathcal{B}_1) - D(\phi^{-1}(B_1^2), \ldots, \phi^{-1}(B_m^2) \mid \mathcal{B}_1) \\ &\geq D(A_1 \mid \mathcal{B}_1) - \sum_{B \in \mathcal{B}_2} D(\phi^{-1}(B) \mid \mathcal{B}_1). \end{aligned}$$

The result also holds with respect to the discrete complexity measures $D_\cap$ and $D_\cup$.

Proof. Let $A_2 = \phi(A_1)$. Since ϕ is injective, $\phi^{-1}(A_2) = A_1$. Let $B_1^2, \ldots, B_m^2, C_1, \ldots, C_t = A_2$ be an extended sequence that describes a construction of A_2 from $\mathcal{B}_2$, where $t = D(A_2 \mid \mathcal{B}_2)$. We claim that

$$\phi^{-1}(B_1^2), \ldots, \phi^{-1}(B_m^2), \phi^{-1}(C_1), \ldots, \phi^{-1}(C_t) = A_1$$

is an extended sequence that describes a construction of A_1 from $\{\phi^{-1}(B_1^2), \ldots, \phi^{-1}(B_m^2)\}$. Indeed, this can be easily verified by induction using that $\phi^{-1}(C_1 \cap C_2) = \phi^{-1}(C_1) \cap \phi^{-1}(C_2)$ and $\phi^{-1}(C_1 \cup C_2) = \phi^{-1}(C_1) \cup \phi^{-1}(C_2)$. The result immediately follows by replacing the initial sets in the construction above by a sequence that realizes $D(\phi^{-1}(B_1^2), \ldots, \phi^{-1}(B_m^2) \mid \mathcal{B}_1)$. □

In particular, if we have a strong enough lower bound with respect to $\langle \Gamma_1, \mathcal{B}_1 \rangle$, and can construct an injective map $\phi \colon \Gamma_1 \to \Gamma_2$ such that for each $B \in \mathcal{B}_2$ the value $D(\phi^{-1}(B) \mid \mathcal{B}_1)$ is small, we get a lower bound in $\langle \Gamma_2, \mathcal{B}_2 \rangle$. Moreover, if the original set A_1 and the map ϕ are "explicit", $A_2 = \phi(A_1)$ is explicit as well.

[7] We note that $\mathsf{vec}(w)$ always has Hamming weight exactly 2 when $\mathcal{B} = \mathcal{G}_{N,M}$ and $w \in [N] \times [M]$. There is a well-known connection between slice functions and graph complexity (see, e.g., [Lok03]).

We provide next a simple example that will be useful later in the text. Given a binary string $w \in \{0,1\}^n$, which we represent as $w = w_1 \dots w_n$, let $\mathsf{number}(w) = \sum_{i=0}^{n-1} 2^i \cdot w_{n-i}$ be the number in $\{0, \dots, 2^n - 1\}$ encoded by w. Let $N = 2^n$, and let $\mathsf{binary}\colon [N] \to \{0,1\}^n$ be the *bijection* that maps the integer $\mathsf{number}(w) + 1$ to the corresponding string $w \in \{0,1\}^n$.

Lemma 13 (Tight transference from graph complexity to circuit complexity)**.** *Let $\langle [N] \times [N], \mathcal{G}_{N,N} \rangle$ and $\langle \{0,1\}^{2n}, \mathcal{B}_{2n} \rangle$ be the discrete spaces corresponding to $N \times N$ graph complexity and $2n$-bit circuit complexity, respectively, where $N = 2^n$. Moreover, let $\phi\colon [N] \times [N] \to \{0,1\}^{2n}$ be the bijective map defined by $\phi(u,v) \stackrel{\text{def}}{=} \mathsf{binary}(u)\mathsf{binary}(v)$. For every $G \subseteq [N] \times [N]$,*

$$D_\cap(\phi(G) \mid \mathcal{B}_{2n}) \;\geq\; D_\cap(G \mid \mathcal{G}_{N,N}).$$

In particular, graph intersection complexity lower bounds yield circuit complexity lower bounds.

Proof. By Lemma 12, it is enough to verify that for each $B \in \mathcal{B}_{2n}$, $D_\cap(\phi^{-1}(B) \mid \mathcal{G}_{N,N}) = 0$. Recall from Section 2.2.1 that $\mathcal{B}_{2n} = \{B_1, \dots, B_{2n}, B_1^c, \dots, B_{2n}^c\}$, where $B_i = \{v \in \{0,1\}^{2n} \mid v_i = 1\}$. if $B_i \in \mathcal{B}_{2n}$ corresponds to the positive literal x_i, then $\phi^{-1}(B_i)$ is either a union of columns (when $i > n$) or a union of rows (when $i \leq n$) in graph complexity (cf. Section 2.2.2). Consequently, in this case $D_\cap(\phi^{-1}(B_i) \mid \mathcal{G}_{N,N}) = 0$ by Facts 4 and 6. On the other hand, for a $B_i^c \in \mathcal{B}_{2n}$, it is not hard to see that $\phi^{-1}(B_i^c)$ also corresponds to either a union of rows or a union of columns. This completes the proof. □

An advantage of Lemma 13 over existing results connecting graph complexity and circuit complexity is that it offers a tighter connection between these two models by focusing on a convenient complexity measure (intersection complexity instead of circuit complexity).[8]

Remark 14 (Circuit lower bounds from graph complexity lower bounds)**.** *Let $C \geq 1$ be a constant. We note that a lower bound of the form $C \cdot \log N$ on $D_\cap(H \mid \mathcal{G}_{N,N})$ for an explicit graph H can be translated into the same lower bound on the circuit complexity of a related explicit Boolean function. In more detail, let $f_H\colon \{0,1\}^{2n} \to \{0,1\}$ be the Boolean function corresponding to a bipartite graph $H \subseteq [N] \times [N]$. Now consider the function $F\colon \{0,1\}^{1+2n} \to \{0,1\}$ defined as follows. The value $F(b,z) = f_H(z)$ if the input bit $b = 1$, and $F(b,z) = \overline{f_H}(z) = 1 - f_H(z)$ if $b = 0$. Note that if H can be computed in time $\mathsf{poly}(N)$ then the corresponding function F is in $E = \mathsf{DTIME}[2^{O(m)}]$, where $m = 2n+1$ is the input length of F. Moreover, if $D_\cap(H \mid \mathcal{G}_{N,N}) \geq C \cdot \log N$ then any Boolean circuit computing F must contain at least $C \cdot 2n$ AND and OR gates in total (assuming a circuit model with access to input literals and without NOT gates). This follows from Lemma 13 and Boolean duality, i.e., that the AND complexity of a Boolean function coincides with the OR complexity of its negation. Formally, letting $\mathcal{B}_\ell$ denote the standard set of generators in the Boolean circuit complexity of ℓ-bit Boolean functions, we have:*

$$\begin{aligned}
D(F \mid \mathcal{B}_m) &\geq D_\cap(F \mid \mathcal{B}_m) + D_\cup(F \mid \mathcal{B}_m) \\
&\geq D_\cap(f_H \mid \mathcal{B}_m) + D_\cup(\overline{f_H} \mid \mathcal{B}_m) - O(1) \\
&= D_\cap(f_H \mid \mathcal{B}_m) + D_\cap(f_H \mid \mathcal{B}_m) - O(1) \\
&\geq 2 \cdot D_\cap(H \mid \mathcal{G}_{N,N}) - O(1) \\
&\geq 2 \cdot C \cdot \log N = C \cdot 2n = C \cdot m - O(1).
\end{aligned}$$

[8] In the Magnification Lemma of [Juk13], it is already implicitly shown that $D_\cap(f_G \mid \mathcal{B}_{2n}) \geq D_\cap(G \mid \mathcal{G}_{N,N})$. However, the literature in graph complexity focuses on the relationship between $D(f_G \mid \mathcal{B}_{2n})$ and $D(G \mid \mathcal{G}_{N,N})$, where there is a constant factor loss. In particular, the best transference bound known is $D(f_G \mid \mathcal{B}_{2n}) \geq D(G \mid \mathcal{G}_{N,N}) - (4 + o(1))N$ (see [Juk13], citing [Cha94]). This means that only a $\Omega(N)$ lower bound on $D(G \mid \mathcal{G}_{N,N})$ would imply a meaningful bound on $D(f_G \mid \mathcal{B}_{2n})$, whereas our setting allows us to transfer a $(1+\varepsilon)\log N$ graph complexity lower bound into a $(1+\varepsilon)n$ circuit lower bound.

Remark 15 (Graph complexity lower bounds from circuit complexity lower bounds)**.** *It is not hard to show by Lemma 12 and a similar argument that a lower bound of the form $\omega(2^n \cdot n)$ on the circuit complexity of a function $h\colon \{0,1\}^{2n} \to \{0,1\}$ implies a $\omega(N)$ lower bound in graph complexity, where $N = 2^n$ as usual. On the other hand, note that by a counting argument there exist graphs computed by a single (unbounded fan-in) union whose corresponding $2n$-bit Boolean function has circuit complexity $\Omega(2^n/n)$. In particular, it follows from Lemma 9 that a Boolean function can have exponential intersection complexity, while the corresponding graph has zero intersection complexity.*

2.5 Cyclic Discrete Complexity

We introduce a variant of the complexity measure $D(\cdot \mid \cdot)$ that allows cyclic constructions. Formally, we use $D^{\circlearrowleft}(A \mid \mathcal{B})$ to denote the *cyclic discrete complexity* of A with respect to $\mathcal{B}$, defined as follows. We consider a *syntactic sequence* $I_i, \ldots, I_t$, together with a fixed operation of the form $I_i = K_{i_1} \star_i K_{i_2}$, where $K_{i_1}, K_{i_2} \in \{I_1, \ldots, I_t\} \cup \mathcal{B}$ and $\star_i \in \{\cap, \cup\}$, for each $i \in [t]$. (Notice that we do not require $i_1, i_2 < i$.) The syntactic sequence is viewed as a formal description instead of an actual construction, and it is evaluated as follows. Initially, $I_i^0 \stackrel{\text{def}}{=} \emptyset$ for each $i \in [t]$. Then, for every $j > 0$, $I_i^j \stackrel{\text{def}}{=} I^{j-1} \cup (K_{i_1}^{j-1} \star_i K_{i_2}^{j-1})$, where the sets in $\mathcal{B}$ remain fixed throughout the evaluation. We say that the syntactic sequence generates A from $\mathcal{B}$ if there exists $j \in \mathbb{N}$ such that $I_t^{j'} = A$ for every $j' \geq j$. Finally, we let $D^{\circlearrowleft}(A \mid \mathcal{B})$ denote the minimum length t of such a sequence, if it exists. The complexity measure $D_{\cap}^{\circlearrowleft}$ is defined analogously, and only takes into account the number of intersection operations in the definition of the syntactic sequence.

Lemma 16 (Convergence of the evaluation procedure)**.** *Suppose $I_1, \ldots, I_t$ together with the corresponding $\star_i$ operations define a syntactic sequence. Then, for every $j \geq t$,*

$$I_i^{j+1} \;=\; I_i^j.$$

In other words, the evaluation converges after at most t steps.

Proof. The evaluation is monotone, in the sense that an element $v \in \Gamma$ added to a set during the j-th step of the evaluation cannot be removed in subsequent updates. From the point of view of this fixed element, if it is not added to a new set during an update, it won't be added to new sets in subsequent updates. Consequently, each set in the sequence converges after at most t iterations. ☐

Corollary 17 (Cyclic discrete complexity versus discrete complexity)**.** *For every set $A \subseteq \Gamma$ and family $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$ of generators,*

$$D_{\cap}^{\circlearrowleft}(A \mid \mathcal{B}) \;\leq\; D_{\cap}(A \mid \mathcal{B}) \;\leq\; D_{\cap}^{\circlearrowleft}(A \mid \mathcal{B})^2.$$

Proof. For the first inequality, observe that from every construction of A from $\mathcal{B}$ we can define an acyclic syntactic sequence that generates A from $\mathcal{B}$. For the second inequality, simply unfold the evaluation of the syntactic sequence into a sequence that generates A from $\mathcal{B}$. Since the additional union operations coming from the update step $I_i^j = I^{j-1} \cup (K_{i_1}^{j-1} \star_i K_{i_2}^{j-1})$ do not increase intersection complexity, the claimed upper bound follows from Lemma 16. ☐

We will employ cyclic discrete complexity in Section 3.4 to exactly characterize the power of the fusion method as a framework to lower bound discrete complexity. We finish this section with a concrete example that is relevant in the context of the fusion method (cf. Section 3.3).

Example: The Fusion Problem $\Pi_{\mathcal{R}}$. Let $[m] = \{1, \ldots, m\}$, $Y \subseteq [m]$ be an initial subset of $[m]$, and $\mathcal{R}$ be a *fixed* set of rules encoded by a set of triples of the form (a, b, c), where $a, b, c \in [m]$ are arbitrary. The meaning of a rule (a, b, c) is that the element c should be added to Y in case this set already contains elements

a and b. We let $\Pi_{\mathcal{R}}$ be the following computational problem: Given an arbitrary initial set $Y \subseteq [m]$ as an input instance, is the top element m eventually added to Y? (Observe that this problem is closely related to the GEN Boolean function investigated in [RM99] and related works.)

Note that, for every fixed set $\mathcal{R}$ of rules, $\Pi_{\mathcal{R}}$ can be decided by a cyclic monotone Boolean circuit that contains exactly $|\mathcal{R}|$ fan-in two AND gates. Indeed, it is enough to consider a circuit over input variables $y_1, \ldots, y_m$ that contains three additional layers of gates, described as follows. The first layer contains fan-in two OR gates $f_1, \ldots, f_m$, where each f_i is fed by the input variable y_i and by a corresponding gate h_i in the third layer. Each rule $(a, b, c) \in \mathcal{R}$ gives rise to a fan-in two AND gate $g_{a,b,c}$ in the second layer of the circuit, where $g_{a,b,c} = f_a \wedge f_b$. Finally, in the third layer we have for each $i \in [m]$ a corresponding OR gate h_i, where

$$h_i = \bigvee_{u,v \in [m], (u,v,i) \in \mathcal{R}} g_{u,v,i}.$$

(We stress that *unbounded fan-in* gates are used only to simplify the description of the circuit.) It is easy to see that the gate f_m computes $\Pi_{\mathcal{R}}$ after at most $O(|\mathcal{R}|)$ iterations of the evaluation procedure.

3 Characterizations of Discrete Complexity via Set-Theoretic Fusion

The technique presented in this section can be seen as a set-theoretic formulation of some results from [Raz89] and [Kar93]. The tighter characterization that appears in Section 3.4 is an adaptation of a result from [NM95].

3.1 Definitions and notation

For convenience, let $U \stackrel{\text{def}}{=} A^c = \Gamma \setminus A$, where Γ is the ambient space. We assume from now on that A is *non-trivial*, i.e., both A and A^c are non-empty.

Definition 18 (Semi-filter)**.** *We say that a non-empty family $\mathcal{F} \subseteq \mathcal{P}(U)$ of sets is a* semi-filter *over U if the following hold:*

- (upward closure) *If $U_1 \in \mathcal{F}$ and $U_1 \subseteq U_2 \subseteq U$, then $U_2 \in \mathcal{F}$.*
- (non-trivial) $\emptyset \notin \mathcal{F}$.

Definition 19 (Semi-filter above w)**.** *We say that $\mathcal{F}$ is* above *an element $w \in \Gamma$* (with respect to $\mathcal{B}$ and $U = A^c$) *if the following condition holds. For every $B \in \mathcal{B}$, if $w \in B$ then $B_U \in \mathcal{F}$.*

Figure 2 illustrates Definition 19 in the particularly simple and attractive 2-dimensional framework of graph complexity considered in this work.

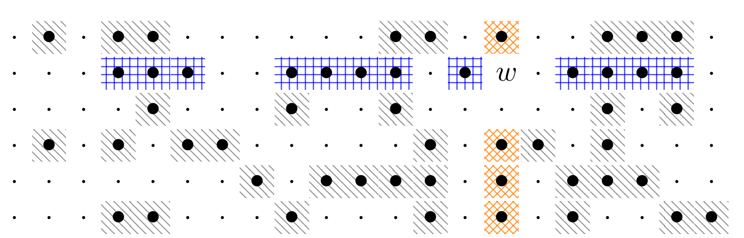

Figure 2: In this example, $\Gamma = [6] \times [22]$, $\mathcal{B} = \mathcal{G}_{6,22}$ (as in Section 2.2.2), and the $\{\cdot, \bullet, w\}$-valued matrix above encodes $U = G^c$ (rectangles with $\bullet$), where $G \subseteq \Gamma$ (locations with $\cdot$ and w) can be interpreted as a bipartite graph. If a semi-filter $\mathcal{F}$ over U is above $w \in G$ (corresponding to coordinates $(2, 15)$), then it must contain the distinguished subsets of U represented in blue $(R_2 \cap U)$ and in orange $(C_{15} \cap U)$, respectively.

Intuitively, semi-filters will be used to produce counter-examples to the correctness of a candidate construction of a set A from $\mathcal{B}$ that is more efficient than $D_\cap(A \mid \mathcal{B})$. This will become clear in Section 3.2.

Definition 20 (Preservation of pairs of subsets)**.** *Let $\Lambda = \{(E_1, H_1), \ldots, (E_\ell, H_\ell)\}$ be a family of pairs of subsets of U. We say that $\mathcal{F}$* preserves *a pair (E_i, H_i) if $E_i \in \mathcal{F}$ and $H_i \in \mathcal{F}$ imply $E_i \cap H_i \in \mathcal{F}$. We say that $\mathcal{F}$* preserves *Λ if it preserves every pair in Λ.*

We now introduce a measure of the *cover complexity* of $A \subseteq \Gamma$ with respect to a family $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$.

Definition 21 (Cover complexity)**.** *We let $\rho(A, \mathcal{B}) \in \mathbb{N} \cup \{\infty\}$ be the minimum size of a collection Λ of pairs of subsets of U such that there is no semi-filter $\mathcal{F}$ over U that preserves Λ and is above an element $a \in A$ (with respect to $\mathcal{B}$ and U).*

The definition of cover complexity considered here is with respect to semi-filters (essentially, non-zero monotone functions). In the context of circuit complexity, notions of cover complexity with respect to other types of Boolean functions (such as ultrafilters and linear functions) have been considered, yielding characterizations of different circuit models [Wig93]. If we ask that in every pair at least one of the sets is the intersection of a generator with U, we obtain characterizations of branching models [Wig95] (such as branching programs). In Section 4.3, we will consider the 2-dimensional cover problem with ultrafilters.

Cover Graph of A and $\mathcal{B}$. In order to get more intuition about the notion of cover complexity, consider an undirected bipartite graph $\Phi_{A,\mathcal{B}} = (V_{\mathsf{pairs}}, V_{\mathsf{filters}}, \mathcal{E})$, where

$$V_{\mathsf{pairs}} \stackrel{\text{def}}{=} \{(E, H) \mid E, H \subseteq U\},$$
$$V_{\mathsf{filters}} \stackrel{\text{def}}{=} \{\mathcal{F} \subseteq \mathcal{P}(U) \mid \mathcal{F} \text{ is a semi-filter and } \mathcal{F} \text{ is above some } a \in A\},$$

and there is an edge $e \in \mathcal{E}$ connecting $(E, H) \in V_{\mathsf{pairs}}$ and $\mathcal{F} \in V_{\mathsf{filters}}$ if and only if $\mathcal{F}$ does not preserve (E, H). Then $\rho(A, \mathcal{B})$ is the minimum number of vertices in V_{pairs} whose adjacent edges cover all the vertices in V_{filters}. For convenience, we say that $\Phi_{A,\mathcal{B}}$ is the *cover graph* of A and $\mathcal{B}$.

Note that a set of vertices in V_{pairs} whose adjacent edges covers all of the vertices in V_{filters} is also known as a *dominating set* in graph theory. Moreover, identifying vertices with their neighbourhoods, the value of $\rho(A, \mathcal{B})$ is equivalent to the optimal value of a set cover problem.

3.2 Discrete complexity lower bounds using the fusion method

Theorem 22 (Fusion lower bound)**.** *Let $A \subseteq \Gamma$ be non-trivial, and $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$ be a non-empty family of generators. Then*

$$\rho(A, \mathcal{B}) \;\leq\; D_\cap(A \mid \mathcal{B}).$$

In other words, the cover complexity of a non-trivial set lower bounds its intersection complexity.

Before proving the result, it is instructive to consider an example. Assume $\Gamma = [N] \times [N]$ and $\mathcal{B} = \mathcal{R}_N$ are instantiated as in Section 2.2.4, where we noted that $D_\cap(G \mid \mathcal{R}_N)$ is always zero. Indeed, $\rho(G, \mathcal{R}_N) = 0$ for every non-trivial $G \subseteq [N] \times [N]$, since in the corresponding cover graph $\Phi_{G,\mathcal{R}_N}$ the vertex set V_{filters} is empty (observe that if a semi-filter is above some $a \in G$, then it must contain the empty set, which is contradictory).

Proof. Let $|\mathcal{B}| = m$ and $D_\cap(A \mid \mathcal{B}) = k$. Assume toward a contradiction that $k < \rho(A, \mathcal{B})$. Let

$$C^1, \ldots, C^m, C^{m+1}, \ldots, C^{m+t} = A \tag{4}$$

be an extended sequence of complexity t that generates A from $\mathcal{B}$, and suppose it has intersection complexity k. Let $U \stackrel{\text{def}}{=} A^c = \Gamma \setminus A$. Recall that, by assumption, both A and U are non-empty. Consider the corresponding relativized sequence

$$C_U^1, \ldots, C_U^m, C_U^{m+1}, \ldots, C_U^{m+t} = \emptyset. \tag{5}$$

This extended sequence generates the empty set from $\mathcal{B}_U$ and has intersection complexity k. Let Λ be the set of intersection operations in this sequence. Note that each pair $(C_U^i, C_U^j) \in \Lambda$ satisfies $C_U^i, C_U^j \subseteq U$, and that $|\Lambda| \leq k < \rho(A, \mathcal{B})$. Let $\Phi_{A,\mathcal{B}} = (V_{\mathsf{pairs}}, V_{\mathsf{filters}}, \mathcal{E})$ be the cover graph of A and $\mathcal{B}$. Since $\Lambda \subseteq V_{\mathsf{pairs}}$ and $|\Lambda| < \rho(A, \mathcal{B})$, there exists $\mathcal{F} \in V_{\mathsf{filters}}$ that is not covered by the pairs in Λ. Let $a \in A$ be an element such that $\mathcal{F}$ is above a.

We trace the construction in Equation 4 from the point of view of the element a. Let $\alpha_i = 1$ if and only if $a \in C_i$. Observe that $\alpha_{m+t} = 1$, since $a \in A$. In order to derive a contradiction, we define a second sequence β_i that depends on the semi-filter $\mathcal{F}$ and on the relativized construction appearing in Equation 5. We let $\beta_i = 1$ if and only if $C_U^i \in \mathcal{F}$ (recall that $\mathcal{F} \subseteq \mathcal{P}(U)$ and $C_U^i \subseteq U$). Since $\mathcal{F}$ is a semi-filter and $C_U^{m+t} = \emptyset$, we get $\beta_{m+t} = 0$. We complete the argument by showing that for every $i \in [m+t]$, $\alpha_i \leq \beta_i$, which is in contradiction to $\alpha_{m+t} = 1$ and $\beta_{m+t} = 0$.

Claim 23. *Suppose $\mathcal{F}$ is above $a \in A$ with respect to $\mathcal{B}$ and U, and that $\mathcal{F}$ preserves Λ, the set of intersection operations in Equation 5. Then for every $i \in [m+t]$, $\alpha_i \leq \beta_i$.*

The proof is by induction on i. For the base case, we consider $i \leq m$. Since $\mathcal{B}$ is non-empty, $m \geq 1$. Now if $\alpha_i = 1$, then $a \in C^i = B$ for some $B \in \mathcal{B}$. Since $\mathcal{F}$ is above a (with respect to $\mathcal{B}$ and U) and $a \in B$, $C_U^i = B_U \in \mathcal{F}$, and consequently $\beta_i = 1$. This completes the base case.

The induction step follows from the induction hypothesis and the upward closure of $\mathcal{F}$ in the case of a union operation, and from the induction hypothesis and the fact that $\mathcal{F}$ preserves Λ in the case of an intersection operation. For instance, suppose that $C^i = C^{i_1} \cap C^{i_2}$ and $C_U^i = C_U^{i_1} \cap C_U^{i_2}$, respectively, where $i_1, i_2 < i$. Assume that $\alpha_i = 1$. Then $a \in C^i$, and consequently $a \in C^{i_1} \cap C^{i_2}$. Using the induction hypothesis, $1 = \alpha_{i_1} = \alpha_{i_2} = \beta_{i_1} = \beta_{i_2}$. Therefore, $C_U^{i_1} \in \mathcal{F}$ and $C_U^{i_2} \in F$. Now using that $(C_U^{i_1}, C_U^{i_2}) \in \Lambda$ and that $\mathcal{F}$ preserves Λ, it follows that $C_U^i = C_U^{i_1} \cap C_U^{i_2} \in F$. In other words, $\beta_i = 1$. The other case is similar.

This establishes the claim and completes the proof of Theorem 22. ☐

3.3 Set-theoretic fusion as a complete framework for lower bounds

In this section, we establish a converse to Theorem 22.

Theorem 24 (Fusion upper bound)**.** *Let $A \subseteq \Gamma$ be non-trivial, and $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$ be a non-empty family of generators. Then*

$$D_\cap(A \mid \mathcal{B}) \;\leq\; \rho(A, \mathcal{B})^2.$$

Remark 25. *It is important in the statements of Theorems 22 and 24 that the characterization of $D_\cap(A \mid \mathcal{B})$ in terms of $\rho(A, \mathcal{B})$ does not suffer a quantitative loss that depends on $|\mathcal{B}|$. This allows us to apply the results in discrete spaces for which the number of generators in $\mathcal{B}$ is large compared to the size of the ambient space Γ, such as in graph complexity.*

Proof. Let $U = A^c$, let $\rho(A, \mathcal{B}) = t$, and assume that this is witnessed by a family

$$\Lambda = \{(H_1, E_1), \ldots, (H_t, E_t)\}$$

of t pairs of subsets of U. We let

$$\mathfrak{F}_\Lambda = \{\mathcal{F} \subseteq \mathcal{P}(U) \mid \mathcal{F} \text{ is a semi-filter that preserves } \Lambda\}.$$

Recall the definition of the cover graph $\Phi_{A,\mathcal{B}}$ of A and $\mathcal{B}$ (Section 3.1). Observe that, while $\Lambda \subseteq V_{\mathsf{pairs}}$, it is not necessarily the case that $\mathfrak{F}_\Lambda \subseteq V_{\mathsf{filters}}$.

Claim 26. *For every $w \in \Gamma$,*

$$w \in A \quad \textit{if and only if} \quad \nexists \mathcal{F} \in \mathfrak{F}_\Lambda \textit{ that is above } w \text{ (}\textit{w.r.t. } \mathcal{B} \textit{ and } U\text{)}.$$

In order to see this, notice that if $w \in A$ then indeed there is no such $\mathcal{F} \in \mathfrak{F}_\Lambda$, using the definitions of ρ and Λ. On the other hand, for $w \notin A$, it is easy to check that $\mathcal{F}_w \stackrel{\text{def}}{=} \{U' \subseteq U \mid w \in U'\}$ is a semi-filter that preserves Λ and that is above w with respect to $\mathcal{B}$ and U.

This claim provides a criterion to determine if an element is in A. This will be used in a construction of A from $\mathcal{B}$ showing that $D_\cap(A \mid \mathcal{B}) = O(\rho(A, \mathcal{B})^2)$. The intuition is that, for a given $w \in \Gamma$, we must check if there is $\mathcal{F} \in \mathfrak{F}_\Lambda$ that is above w with respect to $\mathcal{B}$ and U. In order to achieve this, we inspect the *minimal family* $\mathcal{G}_w \subseteq \mathcal{P}(U)$ of sets that must be contained in any such (candidate) semi-filter.

For every $w \in \Gamma$, we require $\mathcal{G}_w$ to be above w, upward-closed, and to preserve Λ. The rules for constructing $\mathcal{G}_v$ are simple:

- *Base case*. If $w \in B$ for $B \in \mathcal{B}$, then add $B_U = B \cap U$ to $\mathcal{G}_w$, together with every set U' such that $B_U \subseteq U' \subseteq U$.

- *Propagation step*. If both E_i and H_i are in $\mathcal{G}_v$, add $E_i \cap H_i$ to $\mathcal{G}_v$, together with every set U' such that $E_i \cap H_i \subseteq U' \subseteq U$.

We apply the base case once, and repeatedly invoke the propagation step until no new sets are added to $\mathcal{G}_w$. Clearly, this process terminates within a finite number of steps.

Claim 27. *For every $w \in \Gamma$, the empty set is added to $\mathcal{G}_w$ if and only if $w \in A$.*

We argue that $w \notin A$ if and only if $\emptyset \notin \mathcal{G}_w$. Clearly, if $\mathcal{F}$ is a semi-filter that is above w and preserves Λ, we must have $\mathcal{G}_w \subseteq \mathcal{F}$. For $w \notin A$, the process described above cannot possibly add $\emptyset$ to $\mathcal{G}_w$, since by Claim 26 there is a semi-filter $\mathcal{F} \in \mathfrak{F}_\Lambda$ that is above w, and $\mathcal{G}_w \subseteq \mathcal{F}$. On the other hand, if this process terminates without adding $\emptyset$ to $\mathcal{G}_w$, it is easy to see that $\mathcal{G}_w$ is a semi-filter in $\mathfrak{F}_\Lambda$ that is above w, which in turn implies that $w \notin A$ via Claim 26. This completes the proof of Claim 27.

We now turn this discussion into the actual construction of A from the sets in $\mathcal{B}$. For convenience, we actually upper bound $D_\cap(A \mid \mathcal{B} \cup \{\emptyset\})$, i.e., we freely use $\emptyset$ as a generator in the description of the sequence that generates A. This is without loss of generality due to Fact 8. Let

$$\Omega \stackrel{\text{def}}{=} \mathcal{B}_U \cup \{E_i\}_{i\in[t]} \cup \{H_i\}_{i\in[t]} \cup \{H_i \cap E_i\}_{i\in[t]} \cup \{\emptyset\},$$

where we abuse notation and view Ω as a *multi-set*.[9] For simplicity and in order to avoid extra terminology, we slightly abuse notation, and distinguish sets that are identical by the symbols representing them. This should be clear in each context, and the reader should keep in mind what we are simply translating the process that defines each $\mathcal{G}_w$ into a construction of A.

[9] This is helpful in the argument. For instance, more than one set $B \in \mathcal{B}$ might generate an empty set $B_U = B \cap U \in \Omega$, but we will need to keep track of elements such that $w \in B$ and $B_U = \emptyset$.

Fix a set C from the multi-set Ω. For an integer $j \geq 1$, we let S_C^j be the set of all $w \in \Gamma$ that have C in $\mathcal{G}_w$ before the start of the j-th iteration (propagation step) of the process described above. (Here we also view the sets S_C^j as different formal objects.) We construct each set S_C^j from $\mathcal{B} \cup \{\emptyset\}$ by induction on j. By Claim 27, for a large enough $\ell \in \mathbb{N}$, we get $S_\emptyset^\ell = A$, our final goal.

In the base case, i.e., for $j = 1$, we first set $T_{B_U}^1 = B$ for each B_U obtained from a set $B \in \mathcal{B}$, and $T_I^1 = \emptyset$ for every other set $I \in \Gamma$. We then let

$$S_C^1 \;=\; \bigcup_{C' \in \Omega, C' \subseteq C} T_{C'}^1, \tag{6}$$

for each $C \in \Omega$. Observe that the base case satisfies the property in the definition of the sets S_C^j.

Assume we have constructed S_C^{j-1}, for each $C \in \Omega$. We can construct each S_C^j from these sets as follows:

$$T_C^j \;=\; S_C^{j-1} \cup \bigcup_{\{i \in [t] \mid C = E_i \cap H_i\}} (S_{E_i}^{j-1} \cap S_{H_i}^{j-1}), \tag{7}$$

$$S_C^j \;=\; \bigcup_{C' \in \Omega, C' \subseteq C} T_{C'}^j. \tag{8}$$

Note that the definition of each S_C^j handles Λ-preservation and upward-closure, as in the propagation step. It is not difficult to show using the induction hypothesis that each set S_C^j satisfies the required property (fix an element $w \in \Gamma$, and verify that it appears in the correct sets). This completes the construction of A.

In order to finish the proof of Theorem 24, we analyse the complexity of this construction. First, since each propagation step that introduces a new set to $\mathcal{G}_w$ adds at least one of the sets $E_i \cap H_i$ to $\mathcal{G}_w$, and there are at most $t = |\Lambda| = \rho(A, \mathcal{B})$ such sets, it is sufficient in the construction above to take $\ell = t+1$. In particular, $S_\emptyset^{t+1} = A$. Finally, each propagation step (which is associated to a fixed stage $j \in [t]$ of the construction) only employs intersection operations for sets C of the form $E_i \cap H_i$ (in the corresponding definition of T_C^i). Overall, among these sets, the j-th stage of the construction needs at most t intersections. To see this, note that sets S_C^j with $C = E_i \cap H_i$ are only required to inspect the corresponding sets associated with pairs (E_k, H_k) with $k \in [t]$ such that $C = E_k \cap H_k$, and such pairs are disjoint among the different sets C of this form. (There is no need to keep more than one such C representing the same underlying set as a syntactical object in the construction.)

This immediately implies that A can be generated using at most $t(t+1)$ intersections. However, a more careful inspection reveals that the last stage only needs to perform the operations corresponding to the upward-closure, and no new intersections are necessary. Consequently, $D_\cap(A \mid \mathcal{B}) \leq \rho(A, \mathcal{B})^2$, which completes the proof. ☐

We take this opportunity to observe the following immediate consequence of Theorems 22 and 24. (A tighter relation between these measures is discussed in Section 2.3.)

Corollary 28 (Intersection complexity versus discrete complexity)**.**
For every $A \subseteq \Gamma$ and non-empty $\mathcal{B}$, if $D_\cap(A \mid \mathcal{B}) = t$ then $D_\cup(A \mid \mathcal{B}) \leq D(A \mid \mathcal{B}) \leq O(t + |\mathcal{B}|)^3$.

Proof. If A is empty and can be constructed from $\mathcal{B}$, then it can also be constructed from $\mathcal{B}$ using $|\mathcal{B}|$ intersections (and no union operation). If $A = \Gamma$ the same is true with respect to unions. On the other hand, for a non-trivial A, the result follows from Theorems 22 and 24, by noticing that in the construction underlying the proof of Theorem 24 a total of at most $O(t + |\mathcal{B}|)^3$ operations are needed. ☐

Remark 29 (The fusion method and complexity in Boolean algebras)**.** *Our presentation allows us to conclude, in particular, that the fusion method provides a framework to lower bound the number of operations in any* (*finite*) *Boolean algebra* $\mathfrak{B}$*. Indeed, by the Stone Representation Theorem* (*cf.* [GH08])*, any Boolean algebra is isomorphic to a field of sets. Therefore, the problem of determining the number of* $\vee_{\mathcal{B}}$ *and* $\wedge_{\mathcal{B}}$ *operations necessary to obtain a non-trivial element* $a \in \mathfrak{B}$ *from elements* $b_1, \ldots, b_m \in \mathfrak{B}$ *can be captured via cover complexity by Theorems 22 and 24.*

3.4 An exact characterization via cyclic discrete complexity

In this section, we show that cover complexity can be *exactly* characterized using the intersection complexity variant of cyclic complexity. The tight correspondence is obtained by a simple adaptation of an idea from [NM95].

Theorem 30 (Exact characterization of cover complexity)**.** *Let* $A \subseteq \Gamma$ *be non-trivial, and* $\mathcal{B} \subseteq \mathcal{P}(\Gamma)$ *be a non-empty family of generators. Then*

$$\rho(A, \mathcal{B}) \;=\; D^{\circlearrowleft}_{\cap}(A \mid \mathcal{B}).$$

Proof. The proof that $D^{\circlearrowleft}_{\cap}(A \mid \mathcal{B}) \leq \rho(A, \mathcal{B})$ is essentially immediate from the proof of Theorem 24. It is enough to observe that the construction of A from $\mathcal{B}$ via Λ described there can be transformed into a syntactic sequence for A that employs at most $|\Lambda|$ intersection operations. This is similar to the example presented in Section 2.5.

We establish next that $\rho(A, \mathcal{B}) \leq D^{\circlearrowleft}_{\cap}(A \mid \mathcal{B})$. The main difficulty here is that simply unfolding the evaluation of the syntactic sequence introduces further intersection operations (Corollary 17), and we cannot rely on Theorem 22. We argue as follows.

Let $\mathcal{B} = \{B_1, \ldots, B_m\}$, and $I_1, \ldots, I_t$ be a syntactic sequence that generates A from $\mathcal{B}$ using operations $\star_i$, where $t = D^{\circlearrowleft}(A \mid \mathcal{B})$. By Lemma 16, the evaluation procedure converges to a sequence $C^1, \ldots, C^m, C^{m+1}, \ldots, C^{m+t} = A$, where $C^i = B_i$ for $i \in [m]$. (This is not an extended sequence that generates A from $\mathcal{B}$, since the corresponding operations are not acyclic. However, the relation between the sets is clear.)

Claim 31. *If* $I_i = K_{i_1} \star_i K_{i_2}$ *for* $i \in [t]$*, then* $C^j = C^{j'} \diamond_j C^{j''}$*, where* $C^{j'}$ *and* $C^{j''}$ *are the corresponding sets in the sequence above when* $j = i + m$*, and* $\diamond_j \in \{\cap, \cup\}$ *is the corresponding operation.*

In order to see this, recall that during the evaluation of the syntactic sequence $I_i^{\ell+1} = I_i^{\ell} \cup (K_{i_1}^{\ell} \star_i K_{i_2}^{\ell})$. Since the evaluation is monotone, and $C^1, \ldots, C^m, C^{m+1}, \ldots, C^{m+t}$ is the convergent sequence, we eventually have $I_i^{\ell+1} = I_i^{\ell} = (K_{i_1}^{\ell} \star_i K_{i_2}^{\ell})$. Consequently, $C^j = C^{j'} \diamond_j C^{j''}$ after the indices are appropriately renamed.

For $U = A^c$, let $\Lambda \stackrel{\text{def}}{=} \{(C_U^{j'}, C_U^{j''}) \mid j \in \{m+1, \ldots, m+t\} \text{ and } \diamond_j = \cap\}$ be a family of pairs of subsets of U. In order to complete the proof, it is enough to show that Λ covers all semi-filters $\mathcal{F} \subseteq \mathcal{P}(U)$ that are above some element $a = a(\mathcal{F}) \in A$.

Suppose this is not the case, i.e., there is a semi-filter $\mathcal{F}$ above $a \in A$ such that $\mathcal{F}$ is not covered by Λ. We proceed in part as in the proof of Theorem 22. For each $i \in [m+t]$, let $\alpha_i \in \{0, 1\}$ be 1 if and only if $a \in C^i$, and $\beta_i \in \{0, 1\}$ be 1 if and only if $C_U^i \in \mathcal{F}$. We obtain a contradiction by a slightly different argument, which is in analogy to the proof in [NM95]. Since the operations performed over $C^1, \ldots, C^m, C^{m+1}, \ldots, C^{m+t}$ do not follow a linear order, and these sets are obtained after the convergence of the evaluation procedure, we employ a top-down approach, as opposed to the bottom-up presentation that appears in the proof of Theorem 22.

We define a partition (X, Y) of the indices of the sets $C^1, \ldots, C^{m+t}$. Note that $\alpha_{m+t} = 1$ and $\beta_{m+t} = 0$ (cf. Theorem 22). Initially, X contains only the element $m + t$. Now for each $j \in X$, if $C^j = C^{j'} \diamond_j C^{j''}$, $\alpha_{j'} = 1$, and $\beta_{j'} = 0$, then we add the element j' to X (and similarly for the index j''). We repeat this procedure until no more elements are added to X, and let $Y \stackrel{\text{def}}{=} [m + t] \setminus X$.

We observe the following properties of this partition.

Claim 32. *We have $m + t \in X$ and $\{1, \ldots, m\} \subseteq Y$. If an element $j \in X$, then $\alpha_j = 1$ and $\beta_j = 0$.*

The only non-trivial statement is that $\{1, \ldots, m\} \subseteq Y$. It is enough to argue that if $\ell \in [m]$ then it is not the case that $\alpha_\ell = 1$ and $\beta_\ell = 0$. But since $C^\ell = B_\ell \in \mathcal{B}$ and $\mathcal{F}$ is above a, if $\alpha = 1$ (i.e., $a \in C^\ell$) then $\beta = 1$ (i.e., $B_\ell \cap U \in \mathcal{F}$).

Claim 33. *If $j \in X$ and $C^j = C^{j'} \diamond_j C^{j''}$, where $\diamond_j \in \{\cap, \cup\}$ is arbitrary, then either $j' \in X$ or $j'' \in X$.*

Assume contrariwise that $j \in X$ and $j', j'' \in Y$. First, suppose that $\diamond_j = \cap$. Since $\alpha_j = 1$ and $C^j = C^{j'} \cap C^{j''}$, we have $\alpha_{j'} = \alpha_{j''} = 1$. As $j', j'' \in Y$, by construction, we get $\beta_{j'} = \beta_{j''} = 1$ (otherwise one of the indices would be in X and not in Y). Consequently, by the definition of the sequence β, $C^j_U \notin \mathcal{F}$, while $C^{j'}_U, C^{j''}_U \in \mathcal{F}$. This contradictions the assumption that Λ does not cover $\mathcal{F}$. Assume now that $\diamond_j = \cup$. Moreover, suppose w.l.o.g. that $\alpha_{j'} = 1$, which can be done thanks to $C^j = C^{j'} \cup C^{j''}$ and $\alpha_j = 1$. Since $j' \in Y$, we must have $\beta_{j'} = 1$. This means that $C^{j'}_U \in \mathcal{F}$, and by the monotonicity of $\mathcal{F}$ and $\diamond_j = \cup$, it follows that $C^j_U \in \mathcal{F}$. But this is in contradiction to $\beta_j = 0$, which completes the proof of the claim.

Claim 34. *Suppose that $j, j' \in X$, $C^j = C^{j'} \cup C^{j''}$, and $j'' \in Y$. Then $a \notin C^{j''}$.*

The assumptions force $\alpha_j = 1$ and $\beta_j = 0$, and that it is not the case that $\alpha_{j''} = 1$ and $\beta_{j''} = 0$. We must argue that $\alpha_{j''} = 0$ (i.e., $a \notin C^{j''}$), and to do so we show that $\beta_{j''} = 0$. But if $\beta_{j''} = 1$, the monotonicity of $\mathcal{F}$ and $C^j = C^{j'} \cup C^{j''}$ imply $\beta_j = 1$, a contradiction. This completes the proof of this claim.

Finally, we combine these three claims, derived from the assumption that there is a semi-filter $\mathcal{F}$ above a that is not covered by Λ, to get a contradiction. Recall that $C^1, \ldots, C^{m+t} = A$ is the convergent sequence obtained from the syntactic sequence $I_1, \ldots, I_t$ and its operations $\star_i$, and that by assumption $a \in A$. Therefore, our proof will be complete if we can show that $a \notin C^{m+t}$.

In order to establish this final implication, we show the stronger statement that the element a is never added to a set C^j during the update steps of the evaluation procedure if $j \in X$ (since $m + t \in X$ by Claim 32), which is a contradiction. Before the first update, each such set is empty, as the only non-empty sets are in $\mathcal{B}$, and these have indices in Y (Claim 32). During an update of the elements of a set C^j with $j \in X$, we consider two cases based on $\diamond_j \in \{\cup, \cap\}$. If $\diamond_j = \cap$, Claim 33 implies that at least one of the operands comes from X, and thus by induction the update step will not include a in C^j. On the other hand, if $\diamond_j = \cup$, Claim 33 shows that at most one operand comes from Y. If there is no operand from Y, we are done using the induction hypothesis. Otherwise, Claim 34 implies that a is not an element of this operand (as it is not in the corresponding set even after the evaluation procedure converges). By the induction hypothesis, a is not added to C^j. This finishes the proof of Theorem 30. □

In particular, this result shows that the k-clique lower bound discussed in [Kar93] holds in the more general model of cyclic Boolean circuits.

Corollary 35 (Consequence of Theorem 30 and [Kar93])**.**
Let k-clique: $\{0,1\}^{\binom{n}{2}} \to \{0,1\}$ be the function that evaluates to 1 on an undirected n-vertex input graph G if and only if G contains a k-clique. Then every monotone cyclic Boolean circuit that computes 3-clique contains at least $\Omega(n^3/(\log n)^4)$ fan-in two AND *gates.*

This lower bound against monotone cyclic circuits does not seem to easily follow from the proofs in [Raz85, AB87].

4 Graph Complexity and Two-Dimensional Cover Problems

4.1 Basic results and connections

Proposition 36 (The intersection complexity of a random graph)**.** *Let $G \subseteq_{1/2} [N] \times [N]$ be a random bipartite graph. Then, asymptotically almost surely,*

$$D_{\cap}(G \mid \mathcal{G}_{N,N}) \;=\; \Theta(N).$$

Proof. The upper bound is easy, and holds in the worst case as well (see Section 2.2.2). For the lower bound, recall that a random graph G satisfies $D(G \mid \mathcal{G}_{N,N}) = \Omega(N^2/\log N)$, which is an immediate consequence of Lemma 11. By Lemma 9, it must be the case that $D_{\cap}(G \mid \mathcal{G}_{N,N}) = \Omega(N)$, which completes the proof. □

Recall the definition of cover complexity introduced in Section 3.1. Theorem 24 and Proposition 36 yield an $\Omega(\sqrt{N})$ lower bound on the cover complexity of a random graph. It is possible to obtain a tight lower bound using a more careful argument.

Theorem 37 (The cover complexity of a random graph)**.** *Let $G \subseteq_{1/2} [N] \times [N]$ be a random bipartite graph. Then, asymptotically almost surely,*

$$\rho(G, \mathcal{G}_{N,N}) \;=\; \Theta(N).$$

Proof. The proof is based on a counting argument, and can be formalized using Kolmogorov complexity. Observe that the proof of Theorem 24 describes a *universal procedure* that generates an *arbitrary* set A from $\mathcal{B}$ using Λ. However, for a *fixed* family $\mathcal{B}$ such as $\mathcal{B} = \mathcal{G}_{N,N}$, the only information the procedure needs is the inclusion relation among the sets appearing in Λ and $\mathcal{B}$. Crucially, the explicit description of the sets that appear in Λ is not necessary to fully specify the corresponding set A that is generated by the universal procedure. Indeed, observe that the core of the construction after the base case (which does not depend on A) are the sub-indices appearing in Equations 6, 7, and 8, which are determined by the aforementioned inclusion relations. These inclusions can be described by $O(|\Lambda|(|\mathcal{B}| + |\Lambda|))$ bits. Since a random graph has description complexity $\Omega(N^2)$ and $|\mathcal{G}_{N,N}| = 2N$, we must have $|\Lambda| = \Omega(N)$ asymptotically almost surely. In other words, $\rho(G, \mathcal{G}_{N,N}) = \Omega(N)$ for a typical graph $G \subseteq [N] \times [N]$. □

Let $N = 2^n$. For a graph $G \subseteq [N] \times [N]$, we let $f_G \colon \{0,1\}^{2n} \to \{0,1\}$ be the Boolean function associated with G, as described in Lemma 13 (in other words, $f_G^{-1}(1) = \phi(G)$).

Proposition 38 (Reducing circuit complexity lower bounds to two-dimensional cover problems)**.** *For any non-trivial graph $G \subseteq [N] \times [N]$,*

$$\rho(G, \mathcal{G}_{N,N}) \;\leq\; D_{\cap}(f_G^{-1}(1) \mid \mathcal{B}_{2n}).$$

Proof. This follows from Theorem 22 and Lemma 13. □

These results do not immediately imply that $\rho(G, \mathcal{G}_{N,N}) \leq \rho(f_G^{-1}(1), \mathcal{B}_{2n})$, since the connection between $D_{\cap}$ and ρ might not be tight. This can be shown by a direct argument.

Lemma 39 (A fusion transference lemma)**.** *Let $G \subseteq [N] \times [N]$ be a non-trivial graph. Then,*

$$\rho(G, \mathcal{G}_{N,N}) \;\leq\; \rho(f_G^{-1}(1), \mathcal{B}_{2n}).$$

Proof. Let $\mathfrak{F}^{\uparrow}_{f_G}$ be the set that contains a semi-filter $\mathcal{F}$ over $f_G^{-1}(0)$ if and only if it is above some element $a \in f_G^{-1}(1)$. Similarly, let $\mathfrak{F}^{\uparrow}_{G}$ contain a semi-filter $\mathcal{F}$ over $\overline{G}$ if and only if there is $(u,v) \in G$ such that $\mathcal{F}$ is above (u,v). Assume Λ_{f_G} is a family of pairs of subsets of $f_G^{-1}(0)$ that cover all semi-filters in $\mathfrak{F}^{\uparrow}_{f_G}$. Now let Λ_G be the family of pairs of subsets of $\overline{G}$ induced by the pairs in Λ_{f_G} and the bijection between $[N] \times [N]$ and $\{0,1\}^{2n}$. We claim that Λ_G covers all semi-filters in $\mathfrak{F}^{\uparrow}_{G}$.[10]

Recall that we identify an element $(u,v) \in [N] \times [N]$ with its corresponding input string $\phi(u,v) = \mathsf{binary}(u)\mathsf{binary}(v) \in \{0,1\}^{2n}$, which for convenience we will simply denote by uv. Assume this is not the case, i.e., there is a semi-filter $\mathcal{F} \in \mathfrak{F}^{\uparrow}_{G}$ that is above some edge $(u,v) \in G$ and preserves Λ_G (in other words, it is not covered by Λ_G). Let $\mathcal{F}'$ be the corresponding family of subsets of $f_G^{-1}(0)$ under ϕ. Observe that $\mathcal{F}'$ is a semi-filter over $f_G^{-1}(0)$, and that it preserves Λ_{f_G}. Therefore, in order to get a contradiction it is enough to verify that $\mathcal{F}'$ is above uv (with respect to the family of generators $\mathfrak{B}_{2n} \subseteq \mathcal{P}(\{0,1\}^{2n})$). This follows easily using the upward-closure of $\mathcal{F}$ and the fact that $\mathcal{F}$ is above the edge (u,v) with respect to $\mathcal{G}_{N,N}$, as we explain next.

For instance, assume that $u_i = 0$ for some $i \in [n]$. We must prove that the corresponding set $B_i^c \cap f_G^{-1}(0) \in \mathcal{F}'$. From $u_i = 0$, we get $R_u \subseteq \phi^{-1}(B_i^c)$, and then $R_u \cap \overline{G} \subseteq \phi^{-1}(B_i^c) \cap \overline{G} = \phi^{-1}(B_i^c \cap f_G^{-1}(0))$. Since $\mathcal{F}$ is above (u,v) with respect to $\mathcal{G}_{N,N}$, $R_u \cap \overline{G} \in \mathcal{F}$. Consequently, $\phi(R_u \cap \overline{G}) \in \mathcal{F}'$. Now $\phi(R_u \cap \overline{G}) \subseteq \phi(\phi^{-1}(B_i^c \cap f_G^{-1}(0))) = B_i^c \cap f_G^{-1}(0)$, and from the upward-closure of $\mathcal{F}'$, the latter set is in $\mathcal{F}'$. The remaining cases are similar. $\square$

This result and Theorem 22 provide an alternative proof of Proposition 38. As we will see later in this section, establishing a direct connection among cover problems can have further benefits (Section 4.3).

4.2 A simple lower bound example

Let $N = 2^n$. Consider the graph $G_{\mathsf{NEQ}} \subseteq [N] \times [N]$, where $(u,v) \in G_{\mathsf{NEQ}}$ if and only if $u \neq v$. Figure 3 below describes the $N = 8$ case. We show a tight lower bound on $\rho(G_{\mathsf{NEQ}}, \mathcal{G}_{N,N})$. To prove this result, we focus on a particular set of semi-filters. For convenience, we write $G = G_{\mathsf{NEQ}}$.

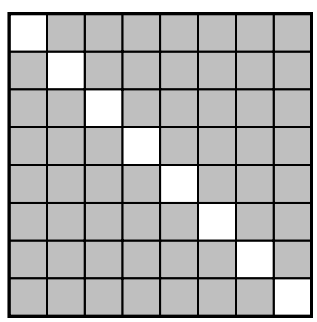

Figure 3: A graphical representation of $G_{\mathsf{NEQ}} \subseteq [N] \times [N]$ for $N = 8$. Proposition 40 shows that for this value of N the intersection complexity is 3.

For $e \in G$, where $e = (u,v)$, we let $\mathcal{F}_e$ be the upward closure (with respect to $\overline{G}$) of the family that contains the sets $R^u_{\overline{G}}$ and $C^v_{\overline{G}}$, where $R^u_{\overline{G}} = R_u \cap \overline{G}$ and $C^v_{\overline{G}} = C_v \cap \overline{G}$. More explicitly, a set W is in $\mathcal{F}_e$ iff $R^u_{\overline{G}} \subseteq W$ *or* $C^v_{\overline{G}} \subseteq W$. Notice that, in general (i.e., for an arbitrary graph), this might not be a semi-filter, as one of the sets might be empty. But for our choice of G, this is a semi-filter above e. We let

$$\mathfrak{F}^{G}_{\mathsf{can}} \stackrel{\text{def}}{=} \{\mathcal{F}_e \mid e \in G \text{ and } \mathcal{F}_e \text{ is a semi-filter }\}.$$

[10]Note that the semi-filters in $\mathfrak{F}^{\uparrow}_{f_G}$ and in $\mathfrak{F}^{\uparrow}_{G}$ differ in their definitions of "above", as they are connected to different sets of generators.

We say that $\mathfrak{F}^G_{\mathsf{can}}$ is the set of *canonical semi-filters* of G (above an edge of G). In general, given a bipartite graph $G \subseteq [N] \times [N]$, how many pairs of subsets of $\overline{G}$ are needed to cover all semi-filters in $\mathfrak{F}^G_{\mathsf{can}}$? Let us denote this quantity by $\rho_{\mathsf{can}}(G, \mathcal{G}_{N,N})$, i.e., the *canonical cover complexity* of G. Clearly, this quantity lower bounds cover complexity.

Proposition 40. *For the graph $G = G_{\mathsf{NEQ}}$ defined above,*

$$\rho_{\mathsf{can}}(G, \mathcal{G}_{N,N}) = \rho(G, \mathcal{G}_{N,N}) = D_{\cap}(G \mid \mathcal{G}_{N,N}) = n = \log N.$$

Proof. The upper bound follows by transforming a circuit for the corresponding Boolean function $f_G \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ into a construction of G. Observe that $f_G(u,v) = \bigvee_{i \in [n]} u_i \oplus v_i$, where $\oplus$ denotes the parity operation, and that each $\oplus$-gate can be implemented using a single $\wedge$-gate via $a \oplus b = (a \vee b) \wedge (\overline{a} \vee \overline{b})$. Therefore, $\rho_{\mathsf{can}}(G, \mathcal{G}_{N,N}) \le \rho(G, \mathcal{G}_{N,N}) \le D_{\cap}(G \mid \mathcal{G}_{N,N}) \le n$ via Lemma 13 and Theorem 22.

For the lower bound on $\rho_{\mathsf{can}}(G, \mathcal{G}_{N,N})$, let $\Lambda = \{(E_1, H_1), \ldots, (E_k, H_k)\}$ be a family of k pairs of subsets of $\overline{G}$. We argue that if Λ covers all semi-filters in $\mathfrak{F}^G_{\mathsf{can}}$ then $k \ge n$. Recall that, for every $e \in G$, $\mathcal{F}_e$ is a semi-filter above e, i.e., $\mathcal{F}_e \in \mathfrak{F}^G_{\mathsf{can}}$. Fix a pair $(E, H) \in \Lambda$.

Claim 41. *Let $e = (u,v) \in G$, and $\mathcal{F}_e \in \mathfrak{F}^G_{\mathsf{can}}$. Then $\mathcal{F}_e$ is covered by (E, H) if and only if each singleton set R^u_G and C^v_G is contained in precisely one of E and H, and none of the latter sets contains both of them.*

Proof of Claim 41. First, we argue that $\mathcal{F}_e$ is covered under the condition in the claim. Assume without loss of generality that $R^u_G \subseteq E$ and $C^v_G \subseteq H$. Then, using the definition of $\mathcal{F}_e$, we get that $E \in \mathcal{F}_e$ and $H \in \mathcal{F}_e$. On the other hand, by assumption, $R^u_G \nsubseteq E \cap H$ and $C^v_G \nsubseteq E \cap H$. This implies that $E \cap H \notin \mathcal{F}_e$. In other words, (E, H) covers $\mathcal{F}_e$.

Suppose now that (E, H) covers $\mathcal{F}_e$. Then $E, H \in \mathcal{F}_e$ but $E \cap H \notin \mathcal{F}$. It is easy to check that this implies the condition in the statement of Claim 41. ☐

Claim 41 immediately implies the following lemma.

Lemma 42. *Every semi-filter in $\mathfrak{F}^G_{\mathsf{can}}$ covered by (E, H) is also covered by $(E \setminus H, H \setminus E)$.*

Thus we can and will assume w.l.o.g. that all pairs appearing in Λ have disjoint sets E_i and H_i. Using Claim 41 again, we obtain the following additional consequence.

Lemma 43. *Every semi-filter in $\mathfrak{F}^G_{\mathsf{can}}$ covered by a disjoint pair (E, H) is also covered by the pair $(E, \overline{G} \setminus E)$.*

Consequently, we will further assume that all pairs appearing in Λ form a partition of $\overline{G}$. Let $(E_1, H_1) \in \Lambda$ be one such pair. Since E_1 and H_1 form a partition of $\overline{G}$, either $|E_1| \ge N/2$ or $|H_1| \ge N/2$. Assume w.l.o.g that $|E_1| \ge N/2$. Let $G_1 \subseteq G$ be the subgraph of G obtained when the ambient space $[N] \times [N]$ is restricted to $\mathsf{Rows}(E_1) \times \mathsf{Columns}(E_1)$, where $\mathsf{Rows}(E_1) = \{a \in [N] \mid (a,b) \in E_1 \text{ for some } b \in [N]\}$, and $\mathsf{Columns}(E_1)$ is defined analogously.

Observe that for no element $e_1 \in G_1$, $\mathcal{F}_{e_1}$ is covered by (E_1, H_1). Furthermore, the elements in G_1 span at least 2^{n-1} different rows and at least 2^{n-1} different columns of $[N]$. Finally, each semi-filter $\mathcal{F}_{e_1} \in \mathfrak{F}^G_{\mathsf{can}}$ for $e_1 \in G_1$ must be covered by some pair in $\Lambda \setminus \{(E_1, H_1)\}$. By a recursive application of the previous argument, and using that in the base case $n = 1$ at least one pair of sets is necessary, it is easy to see $|\Lambda| \ge n = \log N$. This completes the proof. ☐

4.3 Nondeterministic graph complexity

Given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we let $\mathsf{size}(f)$ be the minimum number of fan-in two AND/OR gates in a DeMorgan Boolean circuit computing f (we assume negations appear only at the input level). We can define $\mathsf{size}_\vee(f)$ and $\mathsf{size}_\wedge(f)$ in a similar way. Using our notation, $\mathsf{size}(f) = D(f \mid \mathcal{B}_n)$, $\mathsf{size}_\vee(f) = D_\cup(f \mid \mathcal{B}_n)$, and $\mathsf{size}_\wedge(f) = D_\cap(f \mid \mathcal{B}_n)$.

We also define $\mathsf{conondet\text{-}size}_\wedge(f)$ to be the minimum number of $\wedge$-gates in a circuit $D(x,y)$ such that $f(x) = 1$ if and only if for all y we have $D(x,y) = 1$. Similarly, $\mathsf{nondet\text{-}size}_\vee(g)$ is the minimum number of $\vee$-gates in a circuit $C(x,y)$ such that $g(x) = 1$ if and only if there exists y such that $C(x,y) = 1$. Observe that for every Boolean function h, $\mathsf{conondet\text{-}size}_\wedge(h) = \mathsf{nondet\text{-}size}_\vee(\neg h)$.

Observe that the definition of nondeterministic complexity for Boolean functions relies on Boolean circuits extended with extra input variables. It is not entirely clear how to introduce a natural similar definition in the context of graph complexity, i.e, a nondeterministic version of $D(G \mid \mathcal{G}_{N,N})$. We take a different path, and translate an alternative characterization of nondeterministic complexity in the Boolean function setting (based on the fusion method) to the graph complexity setting. First, we review the necessary concepts.

Definition 44 (Semi-ultra-filter)**.** *We say that a semi-filter $\mathcal{F} \subseteq \mathcal{P}(U)$ is a* semi-ultra-filter *if for every set $A \subseteq U$, at least one of A or $U \setminus A$ is in $\mathcal{F}$.*

For a function $f\colon \{0,1\}^n \to \{0,1\}$, let $\rho_{\mathsf{ultra}}(f, \mathcal{B}_n)$ denote the minimum number of pairs of subsets of $f^{-1}(0)$ that cover all semi-ultra-filters over $f^{-1}(0)$ that are above an input in $f^{-1}(1)$. [Kar93] established the following result.

Theorem 45. *There exists a constant $c \geq 1$ such that for every function $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\rho_{\mathsf{ultra}}(f, \mathcal{B}_n) \;\leq\; \mathsf{conondet\text{-}size}_\wedge(f) \;=\; \mathsf{nondet\text{-}size}_\vee(\neg f) \;\leq\; c \cdot \rho_{\mathsf{ultra}}(f, \mathcal{B}_n).$$

Roughly speaking, a variation of cover complexity can be used to characterize conondeterministic circuit complexity. This motivates the following definition, which provides a notion of nondeterministic complexity in arbitrary discrete spaces.

Definition 46 (Conondeterministic cover complexity)**.** *Given a discrete space $\langle \Gamma, \mathcal{B} \rangle$ and a set $A \subseteq \Gamma$, we let $\rho_{\mathsf{ultra}}(A, \mathcal{B})$ denote the minimum number of pairs of subsets of $U = A^c = \Gamma \setminus A$ that cover all semi-ultra-filters over U that are above an element $a \in A$.*

Observe that $\rho_{\mathsf{ultra}}(A, \mathcal{B}) \leq \rho(A, \mathcal{B})$, since every semi-ultra-filter is a semi-filter. Conondeterministic cover complexity sheds light into the strength of the simple lower bound argument presented in Section 4.2.

Proposition 47. *Let $G_{\mathsf{NEQ}} \subseteq [N] \times [N]$ be the graph defined in Section 4.2. Then,*

$$\rho_{\mathsf{can}}(G_{\mathsf{NEQ}}, \mathcal{G}_{N,N}) \;\leq\; \rho_{\mathsf{ultra}}(G_{\mathsf{NEQ}}, \mathcal{G}_{N,N}).$$

Proof. For convenience, let $G = G_{\mathsf{NEQ}}$. Simply observe that every semi-filter $\mathcal{F}_e$ in $\mathfrak{F}^G_{\mathsf{can}}$ is a semi-ultra-filter. Indeed, for $e = (u,v) \in G$ and an arbitrary set $W \subseteq \overline{G}$, either W or $\overline{G} \setminus W$ contains $R^u_{\overline{G}}$, since the latter is a singleton set due to our choice of G. □

Now we translate this result into a stronger lower bound in Boolean function complexity. This will be a consequence of the following lemma.

Lemma 48 (A nondeterministic fusion transference lemma)**.**
Let $N = 2^n$. For every graph $G \subseteq [N] \times [N]$,

$$\rho_{\mathsf{ultra}}(G, \mathcal{G}_{N,N}) \;\leq\; \rho_{\mathsf{ultra}}(f_G, \mathcal{B}_{2n}),$$

where $f\colon \{0,1\}^{2n} \to \{0,1\}$ is the Boolean function associated with G.

Proof. Recall that, in the proof of Lemma 39 (fusion transference lemma), if a semi-filter $\mathcal{F}$ in the graph setting is not covered, then it gives rise to a semi-filter $\mathcal{F}'$ in the Boolean function setting that is not covered. Crucially, if the original semi-filter is a semi-ultra-filter, so is the resulting semi-filter. The proof of this fact is obvious, since $\phi\colon [N] \times [N] \to \{0,1\}^{2n}$ is a bijection. □

Let $\mathsf{NEQ}_{2n}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be the function such that $\mathsf{NEQ}_{2n}(x,y) = 1$ if and only if $x \neq y$, and EQ_{2n} be its negation. By combining the ideas of this section and Section 4.2, we get the following tight inequalities.

Corollary 49 (A simple nondeterministic lower bound via graph complexity + fusion)**.**

$$\begin{aligned} n &\leq \rho_{\mathsf{can}}(G_{\mathsf{NEQ}}, \mathcal{G}_{N,N}) \\ &\leq \rho_{\mathsf{ultra}}(G_{\mathsf{NEQ}}, \mathcal{G}_{N,N}) \\ &\leq \rho_{\mathsf{ultra}}(\mathsf{NEQ}_{2n}, \mathcal{B}_{2n}) \\ &\leq \mathsf{conondet\text{-}size}_{\wedge}(\mathsf{NEQ}_{2n}) \\ &\leq \mathsf{nondet\text{-}size}_{\vee}(\mathsf{EQ}_{2n}) \\ &\leq \mathsf{size}_{\vee}(\mathsf{EQ}_{2n}) \\ &\leq \mathsf{size}_{\wedge}(\mathsf{NEQ}_{2n}) \\ &\leq n. \end{aligned}$$

In particular, the nondeterministic union complexity of the Boolean function EQ_{2n} *is precisely* n*.*

Observe that, by Theorem 30, a cyclic circuit computing NEQ_{2n} also requires n fan-in two AND gates.

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[Cha94] A. V. Chashkin. On the complexity of boolean matrices, graphs, and the boolean functions corresponding to them. *Discrete Mathematics and Applications*, 4(3):229–258, 1994.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In *Symposium on Foundations of Computer Science* (FOCS), pages 89–98, 2016.

[GH08] Steven Givant and Paul Halmos. *Introduction to Boolean algebras*. Springer, 2008.

[GHKK16] Alexander Golovnev, Edward A. Hirsch, Alexander Knop, and Alexander S. Kulikov. On the limits of gate elimination. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 46:1–46:13, 2016.

[Gol18] Alexander Golovnev. Private communication, 2018.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*. Springer, 2012.

[Juk13] Stasys Jukna. Computational complexity of graphs (book chapter). *Advances in Network Complexity, Quantitative and Network Biology*, pages 99–153, 2013.

[Kar93] Mauricio Karchmer. On proving lower bounds for circuit size. In *Structure in Complexity Theory Conference* (CCC), pages 112–118, 1993.

[Lok03] Satyanarayana V. Lokam. Graph complexity and slice functions. *Theory Comput. Syst.*, 36(1):71–88, 2003.

[LY22] Jiatu Li and Tianqi Yang. $3.1n$ - $o(n)$ circuit lower bounds for explicit functions. In *Symposium on Theory of Computing* (STOC), pages 1180–1193, 2022.

[NM95] Katsutoshi Nakayama and Akira Maruoka. Loop circuits and their relation to Razborov's approximation model. *Inf. Comput.*, 119(2):154–159, 1995.

[Oli18] Igor C. Oliveira. Notes on the method of approximations and the emergence of the fusion method. Manuscript (available online), 2018.

[PRS88] Pavel Pudlák, Vojtech Rödl, and Petr Savický. Graph complexity. *Acta Inf.*, 25(5):515–535, 1988.

[Raz85] Alexander A. Razborov. Lower bounds for the monotone complexity of some Boolean functions. *Soviet Math. Doklady*, 31:354–357, 1985.

[Raz89] Alexander A. Razborov. On the method of approximations. In *Symposium on Theory of Computing* (STOC), pages 167–176, 1989.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

[Sch88] Claus-Peter Schnorr. The multiplicative complexity of Boolean functions. In *International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes* (AAECC), pages 45–58, 1988.

[Wig93] Avi Wigderson. The fusion method for lower bounds in circuit complexity. In *Combinatorics, Paul Erdos is Eighty, Bolyai Math. Society*, pages 453–467, 1993.

[Wig95] Avi Wigderson. Lectures on the fusion method and derandomization. *Technical Report*, 1995.

[Zwi96] Uri Zwick. On the number of ANDs versus the number of ORs in monotone Boolean circuits. *Inf. Process. Lett.*, 59(1):29–30, 1996.