THEORY OF COMPUTING, Volume 10 (15), 2014, pp. 389–419
www.theoryofcomputing.org

SPECIAL ISSUE: ANALYSIS OF BOOLEAN FUNCTIONS

Tight Bounds for Monotone Switching Networks via Fourier Analysis

Siu Man Chan* Aaron Potechin†

Received July 21, 2012; Revised January 22, 2013; Published November 4, 2014

Abstract: We prove tight size bounds on monotone switching networks for the NP-complete problem of k-clique, and for an explicit monotone problem by analyzing a pyramid structure of height h for the P-complete problem of generation. This gives alternative proofs of the separations of m-NC from m-P and of m-NC^i from m-NC^{i+1}, different from Raz–McKenzie (Combinatorica 1999). The enumerative-combinatorial and Fourier analytic techniques in this paper are very different from a large body of work on circuit depth lower bounds, and may be of independent interest.

ACM Classification: F.1.3

AMS Classification: 68Q17, 68Q15, 68Q10

Key words and phrases: lower bounds, space complexity, parallel complexity, monotone complexity, switching networks, Fourier analysis

1 Introduction

To study parallel time and memory usage complexity, lower bounds are sought in different models of computation. For example, parallel time is captured by the depth of Boolean circuits of bounded fan-in,

An earlier version of this paper appeared in the Proceedings of the 44th ACM Symp. on Theory of Computing, pages 495–504, 2011.

*This material is based upon work supported by the National Science Foundation under Grant No. CCF-1017403 and Grant No. CCF-0830797.

†This material is based on work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 0645960.

 DOI: 10.4086/toc.2014.v010a015

and memory usage (i. e., space complexity) is captured by the size of switching networks. It is therefore of interest to prove lower bounds for the depth of Boolean circuits of bounded fan-in, and for the size of switching networks, for an explicit Boolean function. Unfortunately, lower bounds that separate complexity classes have not been proved without certain restrictions on the computation.

To prove some lower bounds, researchers often restrict computation to be *monotone* (i. e., to disallow logical negations in computation) when computing monotone Boolean functions. Improving on Razborov [48], Alon and Boppana [3] and Haken [26] proved that the clique problem requires polynomial ($n^{\Omega(1)}$) depth for monotone circuits.[1] In terms of complexity classes, it says

$$\mathsf{m\text{-}NC} \subseteq \mathsf{m\text{-}P} \subsetneq \mathsf{m\text{-}NP}\,.$$

This means the clique problem requires high parallel time when computed in a monotone fashion. As for the parallel time of efficiently computable functions, Karchmer and Wigderson [34] showed that the directed connectivity problem requires super-logarithmic ($\Omega(\log^2 n)$) depth for monotone circuits,[1] implying

$$\mathsf{m\text{-}NC}^1 \subsetneq \mathsf{m\text{-}NL} \subseteq \mathsf{m\text{-}NC}^2\,.$$

For more separations, Raz and McKenzie [46] extended the lower bound framework of Karchmer and Wigderson [34], proving that a monotone circuit[1] computing

(1) the complete problem for NL, directed connectivity, requires $\Omega(\log^2 n)$ depth, reproving the tight bound of Karchmer and Wigderson [34];

(2) the "complete problem for NC^i," the Generation problem with a pyramid structure of height h, requires $\Omega(h \log n)$ depth when $h \le n^c$ for some constant $c > 0$, giving $\mathsf{m\text{-}NC} \neq \mathsf{m\text{-}P}$ and $\mathsf{m\text{-}NC}^i \neq \mathsf{m\text{-}NC}^{i+1}$ for all i by setting $h = \log^i n$; and

(3) the complete problem for NP, the k-clique problem, requires $\Omega(k \log n)$ depth when $k \le n^c$ for some constamt $c > 0$, improving previous results on cliques for $\log n \ll k \ll n^c$.

Much less work has been done on switching networks, which is a combinatorial model capturing deterministic space-bounded computation (see Section 2 for a discussion). Most results are derived using the connection between circuits and switching networks. Namely, a circuit[1] of depth d can be simulated by a switching network of size 2^d, while a switching network of size s can be simulated by a circuit of depth $O(\log^2 s)$ [16]. As far as lower bounds go, a bound of $\Omega(s)$ for switching network size translates to a bound of $\Omega(\log s)$ for circuit depth, and a bound of $\Omega(d)$ for circuit depth translates to a bound of $2^{\Omega(\sqrt{d})}$ for switching network size. The simulations preserve monotonicity, therefore, so do the translations of lower bounds.

In particular, from the best lower bounds for the depth of monotone circuits by Raz and McKenzie [46], it follows that a monotone switching network computing

(1') directed connectivity requires $n^{\Omega(1)}$ size;

(2') the Generation problem with a pyramid structure of height h requires $n^{\Omega\left(\sqrt{h/\log n}\right)}$ size when $h \le n^c$ for some constant $c > 0$; and

[1]In this paper, we consider only *Boolean* circuits of *fan-in at most two*, even when we do not spell it out.

(3') the k-clique problem requires $n^{\Omega\left(\sqrt{k/\log n}\right)}$ size when $k \le n^c$ for some constant $c > 0$.

Since a size bound of $s^{\Omega(1)}$ for switching networks is equivalent to a space bound of $\Omega(\log s)$ for Turing machines,[2] item (1') above is trivial and fails to separate m-L from m-NL for space complexity on the switching network model.[3] Also, items (2') and (3') are not effective when, for example, h and k are $O(\log n)$. To match our intuition, we expect that a monotone switching network computing

(1") directed connectivity requires $n^{\Omega(\log n)}$ size;

(2") the Generation problem with a pyramid structure of height h requires $n^{\Omega(h)}$ size for $h \le n^c$ for some constant $c > 0$;

(3") k-clique requires $n^{\Omega(k)}$ for $k \le n^c$ for some constant $c > 0$.

Note that such bounds for monotone switching networks would in particular imply the above bounds for monotone circuits by the simulation argument (e. g., (1") implies (1)).

Recently, Potechin [45] indeed strengthened item (1) to item (1"), i. e., showed that any monotone switching network solving directed connectivity has quasi-polynomial size, thereby giving m-L $\neq$ m-NL "on monotone switching networks."[3] To avoid the loss in translation (e. g., from (1) to (1'), as opposed to (1")), it is necessary to depart from the circuit-depth lower bound framework of the communication game by Karchmer and Wigderson [34],[4] which is the basis of most previous work on depth complexity of circuits [15, 19, 22, 25, 29, 33, 34, 46, 47]. Instead, Potechin [45] introduced a Fourier analytic framework for analyzing extremal instances (minterms and maxterms) of the monotone Boolean function of directed connectivity.

1.1 Our results

This work extends the Fourier analytic framework [45] to prove tight lower bounds, for (i) an explicit monotone problem by analyzing a pyramid structure of height h for the P-complete Generation problem, and for (ii) the NP-complete k-clique problem,[5] and as a result strengthens item (2) to item (2"), and item (3) to item (3"). This strengthens the previous bounds in items (2') and (3') for that were translated from items (2) and (3).

[2] Modulo uniformity, of course. The easy direction is folklore (e. g., see [45, §2]), and the hard direction is proved by Reingold [54].

[3] It should be noted that there are at least two combinatorial models for (non-uniform) m-L in the literature: as monotone (Boolean) circuits (of bounded fan-in) of logarithmic *width* and polynomial size [24, 25], or as monotone switching networks of polynomial size [45, 52]. It appears that the two models are not comparable. This work focuses on monotone switching networks of polynomial size as the combinatorial model for (non-uniform) m-L, and does *not* imply results for monotone circuits of logarithmic width and polynomial size as (non-uniform) m-L.

[4] Indeed, the Karchmer–Wigderson framework is unlikely to separate m-L from m-NL on monotone switching networks,[3] since it is able to prove the same bound of $\Omega(\log^2 n)$ for the depth of monotone circuits solving *undirected* connectivity [34, 46], which is in L [54] and computable by monotone switching networks of size n^2. This shows that the quadratic relation between circuit-depth and (the logarithm of) switching-network-size in the simulation argument of Borodin [16] is tight.

[5] To see the matching upper bounds, there are (uniform) monotone switching networks for (i) "the problem computed by the universal degree-h reversible pebbling switching network" of size $n^{O(h)}$ (see Section 3.3.5); and (ii) the k-clique problem of size $k^{O(1)} n^{O(k)}$.

In particular, this gives alternative proofs of the separations of monotone complexity classes like $\mathsf{m\text{-}NC} \neq \mathsf{m\text{-}P}$ and $\mathsf{m\text{-}NC}^i \neq \mathsf{m\text{-}NC}^{i+1}$, using very different arguments compared to Raz–McKenzie [46].[6] These monotone separations are necessary for the corresponding non-monotone separations, because a non-monotone separation (e. g., $\mathsf{NC} \neq \mathsf{P}$) implies the corresponding monotone separation (e. g., $\mathsf{m\text{-}NC} \neq \mathsf{m\text{-}P}$).[7] To prove the lower bounds of items (2) and (3), this work simplifies the Fourier analytic framework of Potechin [45] used for analyzing the directed connectivity problem, by studying invariants in the Fourier domain (Lemma 3.44 for the generation problem), and by making explicit the role of the invariants as an inclusion-exclusion principle (see, e. g., Remark 3.39, Lemma 4.14, and Claims 4.16 and 4.17). Therefore, this work provides a combinatorial understanding of the Fourier analytic framework [45]. Perhaps, more previous results may be strengthened and other problems may be studied by extending this Fourier/combinatorial framework, as an alternative or a complement to the Karchmer–Wigderson framework, after three separation results (1), (2) and (3) on monotone circuit-depth are strengthened to (1”), (2”) and (3”).

1.2 Other related work

We did not discuss other work less relevant to the results presented here, but suggesting lower bounds to space and parallel complexity [1, 2, 7–9, 18, 21, 39, 41, 43, 51, 53, 59, 60], or implying lower bounds to monotone depth in general [4, 5, 11, 14, 23, 27, 31, 32, 42, 49, 50, 55–57, 61, 62] or applying monotone depth to study other complexity measures [12, 13, 28, 35, 37]. For more discussion on the complexity of monotone Boolean circuits, the reader is referred to [46, §1], and [15, 24]. For more discussion on switching networks and other models of space-bounded computation, the reader is referred to [52].

We should briefly comment on *branching programs,* another (more popular) combinatorial model for studying deterministic space-bounded computation. Nonuniform space is linearly related to (is in Θ relation with) the logarithm of the size of layered branching program (LBPs), as well as to the logarithm of the size of switching networks (SNs). (The latter statement depends on Reingold’s theorem [54].) It follows that for all Boolean functions, SN size and LBP size are polynomially related. However, no concept of monotone branching programs seems to be known, nor is there an agreed definition of monotone space complexity. Monotone SNs offer a plausible model of (nonuniform) monotone space complexity, hence our choice of the model. An alternative model would be width-bounded monotone circuits; as far as we know, the two models may not be comparable (cf. footnote 3).

1.3 Organization

Section 2 collects notions, notation, and conventions used in the introduction and common to Section 3 and Section 4. Section 3 treats the generation problem, whose lower bound (item (2”)) is proved as Theorem 3.46. Section 4 treats the clique problem, whose lower bound (item (3”)) is proved as Theorem 4.19.

[6] This paper further simplifies the proof for the generation problem (item (2”)) in the STOC’12 version.

[7] And proving monotone lower bounds is in a sense sufficient, since non-monotone separations would follow from the monotone lower bounds of some variants of the problems, see, e. g., [21, §4.1].

2 Preliminaries

Recall the Iverson bracket notation: $[\![condition]\!]$ to mean 1 if *condition* is TRUE, and 0 otherwise. Also, $[n] := \{0,1,\dots,n-1\}$ for $n \in \mathbb{N}$. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, an instance $y \in \{0,1\}^n$ is a YES-instance if $f(y)=1$, and a NO-instance if $f(y)=0$. A Boolean function is *monotone* if $f(x) \le f(y)$ whenever $x \preceq y$, i. e., whenever $x_i \le y_i$ for all $i \in [n]$. For a monotone Boolean function f, a *minterm* y is a $\preceq$-minimal YES-instance, i. e., $f(y)=1$ but $f(x)=0$ for all $x \prec y$; a *maxterm* y is a $\preceq$-maximal NO-instance, i. e., $f(y)=0$ but $f(x)=1$ for all $x \succ y$.

Following Potechin [45], we denote objects of instances (e. g., variable x_i, element u for GEN, vertex v for CLIQUE) by unprimed letters, and objects of switching networks (e. g., node $a' \in V'$, edge $e' \in E'$) primed letters.

Definition 2.1 (Switching Networks)**.** Consider a collection $x_1, x_2, \dots, x_n$ of Boolean variables.

A *switching network* G' has data $(V', E', s', t', \lambda')$, where V' is the set of nodes and E' is the set of edges of an undirected (multi) graph with two distinguished nodes $s', t' \in V'$. Each edge $e' \in E'$ of G' is labeled with a literal x_i or $\bar{x}_i$ with $i \in [n]$, specified by $\lambda'(e')$. A switching network is *monotone* if all edges are labeled with positive literals (for all e', $\lambda'(e') = x_i$ for some i).

An instance $y \in \{0,1\}^n$ is *accepted* by G' if there is a path P' connecting s' and t' using edges labeled with literals in y; namely, $P' =: \langle e'_1, e'_2, \dots, e'_\ell \rangle$, where $e'_j \in E'$ for $1 \le j \le \ell$ connects v'_{j-1} and v'_j, with $v'_0 = s'$ and $v'_\ell = t'$, satisfying (1) if $\lambda'(e'_j) = x_i$ then $y_i = 1$; or (2) $\lambda'(e'_j) = \bar{x}_i$ then $y_i = 0$. Otherwise, y is *rejected* by G'. The Boolean function $f_{G'}$ *computed* by G' is identified with the collection of accepted instances, so $f_{G'}(y) = 1$ iff G' accepts y.

We say that an instance $y \in \{0,1\}^n$ *reaches* a node $a' \in V'$ if there is a path P' connecting s' and a' using edges labeled with literals in y. The undirectedness of a switching network mirrors the reversibility in deterministic space-bounded computation [40,54].[8] Hence switching networks compute by reachability in a reversible way. The *size* of a switching network is the number of edges [52].[9]

We need the following version of Nisan–Wigderson combinatorial design [44]. For a proof and further references, see [58, Lemma 8].

Lemma 2.2 (Combinatorial Design)**.** *For any positive integers q, m, k with $k \le m$, there exist q sets Q_1, $Q_2, \dots, Q_q \subseteq [N]$ where $N := \mathrm{e}^{1+(\ln q)/k} \cdot (m^2/k)$, such that $|Q_i| = m$ for $1 \le i \le q$ and $|Q_i \cap Q_j| \le k$ for $1 \le i < j \le q$.*

3 Lower bound for generation

This section proves the lower bound for (the promise problem of) generation as Theorem 3.46, and uses it in establishing a tight bound for a monotone decision problem as Theorem 3.48. After defining the problem and the model below (Definitions 3.1 and 3.2), Section 3.1 introduces a semantic restriction

[8]For *non*-deterministic space-bounded computation, the corresponding model is called a switching-and-rectifier network (see, e. g., [52]), whose underlying graph can be directed.

[9]All lower bounds in this work concern the number of *nodes*, which is polynomially related to the number of edges here. Also, constants are not optimized.

based on reversible pebbling, and Section 3.2 proves an optimal lower bound for monotone switching networks with this restriction. Section 3.3 proves a lower bound without this restriction, by reducing the general case to the reversible pebbling case.

Definition 3.1 (Generation Problem)**.** For a size parameter n, the *generation problem* (GEN) receives as input a function $e\colon [n]\times[n]\times[n]\to\{0,1\}$. Let $s:=0$ and $t:=n-1$. We think of s as the source and t as the target. We say that s *generates* $v\in[n]$ if (1) v is s; or (2) s generates both w and u, and $e(w,u,v)=1$. GEN problem accepts input e if s generates t. The value of e is represented as n^3 Boolean variables, and the positive literal corresponding to $e(w,u,v)$ is suggestively denoted by $w\wedge u\to v$.

The GEN problem is a monotone variant of the first P-complete problem called PATH SYSTEMS [17]. Subproblems of GEN with additional restrictions on e are complete for smaller complexity classes like non-deterministic logspace (NL) and Nick's class (NC) [6,30]. To specialize monotone switching networks (Definition 2.1) to GEN, we just need to specialize the labeling function λ'.

Definition 3.2 (Monotone Switching Networks for GEN)**.** We say that a switching network is a monotone switching network for GEN (mGEN network), if each edge $e'\in E'$ is labeled with $\lambda'(e')=w\wedge u\to v$ for some $w,u,v\in[n]$. For an instance for GEN with input e, the literal $\lambda'(e')=w\wedge u\to v$ is *in the instance* if $e(w,u,v)=1$.

3.1 Reversible pebbling switching networks for generation

As in [45], we begin by considering a simpler class of mGEN networks which are semantically restricted. In particular, we consider mGEN networks corresponding to the reversible pebble game [10,38], which we call reversible pebbling mGEN networks (Definition 3.6). In this subsection (and henceforth in this paper), focus on triples $w\wedge u\to v$ where $v\neq w$ and $v\neq u$, so that a move allowed by $w\wedge u\to v$ (Remark 3.3) corresponds to a reversible pebble move.

Remark 3.3 (Reversible Pebbling for GEN)**.** In the reversible pebble game for GEN, we start with a pebble on s and try to put a pebble on t. There is only one allowed move:

1. If $e(w,u,v)=1$ and both w and u are pebbled, then we may pebble or unpebble v (i. e., add or remove a pebble on v).[10]

A configuration $K\subseteq[n]$ of the reversible pebble game specifies the elements that are pebbled.

The idea is that each node $a'\in V(G')$ in a reversible pebbling mGEN network G' will correspond to a configuration $K_{a'}$ in the reversible pebble game, and each edge $e'\in E(G')$ in G' with label $\lambda'(e')=w\wedge u\to v$ will correspond to a move in the reversible pebble game which can be done when $e(w,u,v)=1$. However, to make this precise we must first deal with two issues.

The first issue is that the reversible pebble game has many winning configurations (i. e., configurations in which t is pebbled), but a switching network G' has only *one* accepting state t'. To fix this, we modify the reversible pebble game so that all winning configurations are effectively the same.

[10] By contrast, any vertex can be unpebbled at any time in standard black pebbling, even when not all of its immediate predecessors (playing the role of w and u here) are pebbled.

Remark 3.4 (Modified Reversible Pebbling for GEN)**.** In the modified reversible pebble game for GEN, we start with a pebble on s and try to put a pebble on t. There are two allowed moves:

1. If $e(w,u,v) = 1$ and both w and u are pebbled, then we may pebble or unpebble v; and
2. If t is pebbled then we may pebble or unpebble any other element $x \neq t \in [n]$.

The second issue is that if a sequence of moves can be done when $e(w,u,v) = 1$, and that sequence of moves can bring a configuration $K_{a'}$ to another configuration $K_{b'}$, then we may as well add an edge between a' and b' with label $w \wedge u \to v$. That is, it makes more sense for an edge e' in a reversible pebbling mGEN network G' with label $\lambda'(e') = w \wedge u \to v$ to correspond to a sequence of moves (rather than a single move) that can be done if $e(u,w,v) = 1$. This leads to the following definition.

Definition 3.5 (Reversible Pebbling Equivalence)**.** We say that two configurations $K_1, K_2 \subseteq [n]$ are *l-equivalent* for $l := w \wedge u \to v$ if it is possible to bring configuration K_1 to configuration K_2 using a sequence of the following moves:

1. If w and u are pebbled, then we may pebble or unpebble v; and
2. If t is pebbled, then we may pebble or unpebble any other element $x \neq t \in [n]$.

We can now define reversible pebbling mGEN networks.

Definition 3.6 (Reversible Pebbling Networks)**.** We say that an mGEN network G' is a *reversible pebbling mGEN network* if each node $a' \in V'$ can be associated with a reversible pebbling configuration $K_{a'} \subseteq [n]$ satisfying the following conditions:

1. $K_{s'} = \{s\}$ and $K_{t'} = [n]$; and
2. If there is an edge with label $l := w \wedge u \to v$ between nodes a' and b', then $K_{a'}$ and $K_{b'}$ are l-equivalent.

Since it is only possible to win the reversible pebble game when t can be generated from s, every reversible pebbling mGEN network G' is *sound*, i. e., if G' accepts an input e, then e is a YES-instance for GEN. However, it need not compute GEN, since it need not be *complete*, i. e., accepting all YES-instances of GEN.

Before moving on, we give a more convenient characterization of l-equivalence (Proposition 3.9). We can do this because the partial order (under set inclusion $\subseteq$) on configurations of the reversible pebble game for GEN respects l-equivalence, i. e., there is a *unique* maximal configuration in each l-equivalence class (Proposition 3.8).

Definition 3.7 (Maximal Configuration)**.** For a configuration $K \subseteq [n]$ and a triple $l := w \wedge u \to v$, define $K + w \wedge u \to v$ to be the $\subseteq$-maximal configuration in the l-equivalence class containing K.

Proposition 3.8 (Maximal Configuration)**.** *$K + w \wedge u \to v$ is well-defined and*

$$K + w \wedge u \to v = \begin{cases} K & \text{if } t \notin K, \text{ and also } w \notin K \text{ or } u \notin K, \\ K \cup \{v\} & \text{if } t \notin K,\ v \neq t,\ w \in K \text{ and } u \in K, \\ [n] & \text{if } t \in K, \text{ or } v = t \text{ and } w \in K \text{ and } u \in K. \end{cases}$$

Proposition 3.9 (Reversible Pebbling Equivalence)**.** *K_1 and K_2 are l-equivalent for $l := w \wedge u \to v$ if and only if $K_1 + l = K_2 + l$.*

3.2 Pyramid YES-instances and reversible pebbling lower bound

Following Raz and McKenzie [46], consider those YES-instances of GEN having a structure of a pyramid graph.[11] Pyramid YES-instances of GEN are analogous to path YES-instances of directed connectivity in [45]. In particular, they are minterms of the respective monotone Boolean functions, hence "hardest to accept."

Definition 3.10 (Pyramid Graph and YES-Instance)**.** For a parameter h, the *pyramid graph* of height h has $\binom{h+1}{2}$ vertices $V_h := \{v_{r,c}\}_{1 \le c \le r \le h}$. The vertex $v_{r,c}$ is on the r^{th} row, and V_h is laid out row by row with the first row on top and the h^{th} row at the bottom. For $1 \le r \le h-1$, the vertex $v_{r,c}$ has left child $v_{r+1,c}$ and right child $v_{r+1,c+1}$. The pyramid graph is also viewed as a directed graph with arcs pointing from children to their parents (from $v_{r+1,c}$ to $v_{r,c}$ and from $v_{r+1,c+1}$ to $v_{r,c}$).[11]

An instance of GEN forms a *pyramid* YES-*instance* if there exists $Q \subseteq [n] \setminus \{s,t\}$ and a bijective identification $P\colon Q \hookrightarrow\!\!\!\!\rightarrow V_h$ to vertices V_h of a pyramid graph, and $e(w,u,v) = 1$ iff (w,u,v) is a triple in $[n] \times [n] \times [n]$ *consistent* with P, meaning that

1. $w = u = s$ and $v \in Q$ and $P(v)$ is on the bottom row of V_h; or
2. $w = u \in Q$ and $P(w)$ is on the top row of V_h, and $v = t$; or
3. $w, u, v \in Q$ and $P(w)$ and $P(u)$ are different children of $P(v)$.

In this case, the YES-instance is simply called a *pyramid*, and is denoted by $G(P)$.

Note that the identification P specifies a structure of a pyramid graph over $V(P) := Q = P^{-1}(V_h) \subseteq [n] \setminus \{s,t\}$ in $G(P)$. Since a reversible pebbling switching network must "pebble vertices one at a time" (before pebbling t), on any computation path P' accepting a pyramid $G(P)$, there must be a node $b' \in V(P')$ mentioning lots of vertices of $V(P)$ exclusively, i. e., $K_{b'} \setminus \{s\} \subseteq V(P)$ and $|K_{b'} \setminus \{s\}| = h$ (Lemma 3.11).[12] Formally, every path from s' to t' on a reversible pebbling mGEN network gives a reversible pebbling strategy to pebble t, starting from the initial configuration $K_{s'}$ (Remark 3.3). Since a reversible pebbling strategy is also a black pebbling strategy, the lower bound for black pebbling applies also to reversible pebbling.[13]

Lemma 3.11 (Barrier Size Bound)**.** *Fix a pyramid P of height h. Consider a path $P' =: \langle e'_1, e'_2, \ldots, e'_\ell \rangle$ on a reversible pebbling mGEN network, where e'_j connects v'_{j-1} and v'_j and is labeled a positive literal $\lambda'(e'_j) \in G(P)$, from $s' := v'_0$ to $t' := v'_\ell$. (Hence $K_{v'_{j-1}}$ and $K_{v'_j}$ are l_j-equivalent when $l_j := \lambda'(e'_j)$ for $1 \le j \le \ell$.)*

If $K_{s'} = \{s\}$ and $K_{t'} \ni t$, then P' has a node $b' \in V(P')$ with $K_{b'} \setminus \{s\} \subseteq Q$ and $|K_{b'} \setminus \{s\}| = h$.

[11] The pyramid graph is routinely used for studying space complexity, especially using pebbling games [17, 36]. We follow their convention that arcs point from the $(r+1)^{\text{st}}$ row to the r^{th} row.

[12] To prove Theorems 3.12 and 3.13, it suffices for b' to have $|K_{b'} \setminus \{s\}| \ge h$, instead of an exact equality.

[13] The converse (that reversible pebbling number is a lower bound on the black pebbling number) is not true in general. For example, the line graph (i. e., a single path) having ℓ edges requires two pebbles in black pebbling, but it takes $\Theta(\log \ell)$ pebbles in reversible pebbling [45].

Proof. By the proof of the black pebbling lower bound for a pyramid graph [17]. □

Such b' is a *barrier node* for P. Note that if two pyramids share less than h vertices, then they must have different barrier nodes. With this idea, we are ready to prove size lower bounds for reversible pebbling mGEN networks (Theorems 3.12 and 3.13).

Theorem 3.12 (Reversible Pebbling Lower Bound for GEN)**.** *Any reversible pebbling mGEN network G' having n' nodes and computing* GEN *of size n satisfies $n' \geq n^{\Theta(n^{1/10})}$.*

Proof. Note that G' accepts all pyramid YES-instances. Apply Theorem 3.13. □

Theorem 3.13 (Reversible Pebbling Lower Bound for Pyramids)**.** *For any $4 \leq h \leq n^{1/10}$, any reversible pebbling mGEN network G' having n' nodes for* GEN *of size n, and accepting all pyramid* YES*-instances of height h, satisfies $n' \geq n^{h/10} = n^{\Theta(h)}$.*

Proof. Fix h, let $m := \binom{h+1}{2}$, $q := n^{h/10}$, $k := h-1$. Then Lemma 2.2 gives q sets $Q_1, Q_2, \ldots, Q_q \subseteq [N]$ with $|Q_i| = m$ and $|Q_i \cap Q_j| \leq k$ for $1 \leq i \neq j \leq q$, where

$$N \leq \exp\left(\frac{h \ln n}{10(h-1)} + 1\right) h^4 \leq n^{1/3} h^4 \ll n\,.$$

For each Q_i, construct a pyramid P_i by identifying Q_i arbitrarily with a pyramid graph of height h. Now $G(P_i)$ is accepted by G', via some path P_i' labeled with positive literals in $G(P_i)$. Hence Lemma 3.11 gives a node b_i' whose reversible pebbling configuration satisfies $K_{b_i'} \setminus \{s\} \subseteq Q_i$ and $|K_{b_i'} \setminus \{s\}| = h$. Since $|Q_i \cap Q_j| \leq h-1$, we have $b_i' \neq b_j'$ for $1 \leq i \neq j \leq q$. Hence $n' \geq q = n^{h/10}$. □

3.3 Lower bound beyond reversible pebbling

For the general lower bound for mGEN networks, we extend the Fourier analysis framework of [45] (summarized below as Lemma 3.29). Fourier analysis will be done over extremal NO-instances (Definition 3.14) which can be identified with the Fourier basis (Definition 3.17).

It turns out that any mGEN network computing the generation problem must do non-trivial work per individual pyramid, and the work will be visible in the Fourier spectrum. The work done for a pyramid P_1 will be orthogonal to the work done for a different pyramid P_2 if they are well-separated, i. e., P_1 and P_2 share less than h vertices. Since there are $n^{\Theta(h)}$ different pyramids that are pairwise well-separated, an mGEN network must do an amount of work scaling with $n^{\Theta(h)}$, giving the size lower bound.

3.3.1 Fourier analysis of extremal NO-instances

Let $\mathcal{K} := \{K \subseteq [n] \colon K \ni s\}$ be the collection of reversible pebbling configurations containing s. A reversible pebbling configuration $K_{a'}$ is *proper* if $K_{a'} \not\ni t$. Define $\tilde{V} := [n] \setminus \{s,t\}$ and let

$$\mathcal{C} := \mathsf{PowerSet}(\tilde{V}) := \{C \colon C \subseteq \tilde{V}\}$$

be the (bi-)colorings of $\tilde{V}$. Any $C \in \mathcal{C}$ can be associated with a subset of $[n]$ containing s but excluding t by $C \mapsto C \cup \{s\}$. This association gives a bijection between $\mathcal{C}$ and the sub-collection of the reversible pebbling configurations in $\mathcal{K}$ not containing t. While the objects in $\mathcal{C}$ and $\mathcal{K}$ are very different, it will be convenient to compare them via this bijection, i. e., when writing $K_{a'} = C$ or $K_{a'} \subseteq C$, we mean to test for the associated reversible pebbling configuration, e. g., $K_{a'} = C \cup \{s\}$ or $K_{a'} \subseteq C \cup \{s\}$.

Consider extremal NO-instances for GEN that can be identified with some $C \in \mathcal{C}$. Extremal NO-instances of GEN are analogous to cut NO-instances of directed connectivity in [45].[14] In particular, they are maxterms of respective monotone Boolean functions, hence "hardest to reject."

Definition 3.14 (Extremal NO-Instances)**.** Any $C \in \mathcal{C}$ is associated with the extremal NO-instance $G(C)$ where $e(w,u,v) = 0$ iff $w \in C \cup \{s\}$ and $u \in C \cup \{s\}$ and $v \notin C \cup \{s\}$ (recall the association mentioned above). Then the NO-instance $G(C)$ generates $v \in [n]$ iff $v \in C \cup \{s\}$.[15]

Proposition 3.15 (Extremal NO-Instances)**.** *If $K_{a'} + l = K_{b'} + l$ and $l \in G(C)$, then $K_{a'} \subseteq C$ iff $K_{b'} \subseteq C$.*[16]

Proof. If $l \in G(C)$, then $K_{a'} \subseteq C$ iff $K_{a'} + l \subseteq C$. □

The analysis will focus on the collection of extremal NO-instances $G(\mathcal{C}) = \{G(C) : C \in \mathcal{C}\}$.

Definition 3.16 (Inner Product Space of Extremal NO-Instances)**.** A $\mathcal{C}$-vector is a real vector indexed by $\mathcal{C}$, equivalently a function from $\mathcal{C}$ to $\mathbb{R}$. For two $\mathcal{C}$-vectors f and g, we define the inner product

$$\langle f, g \rangle := \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} f(C) g(C)\,,$$

inducing a norm $\|f\| := \langle f, f \rangle^{1/2}$.

Definition 3.17 (Fourier Analysis)**.** Given an extremal NO-instance $U \in \mathcal{C}$, we define the $\mathcal{C}$-vector

$$\chi_U(C) := (-1)^{\sum_{v \in \tilde{V}} [\![v \in U]\!][\![v \in C]\!]}\,.$$

Then the collection $\{\chi_U\}_{U \in \mathcal{C}}$ forms an orthonormal basis of the space of all $\mathcal{C}$-vectors, called the Fourier Basis. For a $\mathcal{C}$-vector g, its Fourier coefficient at $U \in \mathcal{C}$ is $\hat{g}(U) := \langle g, \chi_U \rangle$.

Let $\mathbf{1}$ denote the all-ones $\mathcal{C}$-vector: $\mathbf{1}(C) = 1$ for all $C \in \mathcal{C}$; and similarly $\mathbf{0}$ the all-zeros $\mathcal{C}$-vector. Note that $\chi_{\{\}} = \mathbf{1}$.

3.3.2 Invariant cover for generation

This subsection shows how to use Fourier analysis to obtain size lower bounds on sound mGEN networks (Lemma 3.29). We begin by assigning $\mathcal{C}$-vectors to the nodes of mGEN networks.

[14] Similar considerations of extremal NO-instances appeared also in Raz–McKenzie [46], although the extremal NO-instances they considered are not maxterms in their setting.

[15] This implies that $G(\mathcal{C})$ are all maxterms of GEN. To see this, given any NO-instance G of GEN, consider $C :=$ set of elements (except s) generated by G, then $G \subseteq G(C)$ (when comparing the set of positive literals).

[16] Recall that we write $K_{a'} = C$ and $K_{a'} \subseteq C$ to mean $K_{a'} = C \cup \{s\}$ and $K_{a'} \subseteq C \cup \{s\}$ when comparing a reversible pebbling configuration $K_{a'} \in \mathcal{K}$ with a bi-coloring $C \in \mathcal{C}$.

Definition 3.18 (Annotated mGEN Networks)**.** A *function description* of an mGEN network G' is a function $F : V' \to \mathbb{R}^{\mathcal{C}}$ (i. e., an assignment of a $\mathcal{C}$-vector $F_{a'}$ to each $a' \in V'$). A function description F is *valid* for G' if:

1. $F_{s'} = \mathbf{1}$ and $F_{t'} = \mathbf{0}$; and
2. If a' and b' are connected by an edge labeled l, where $l \in G(C)$, then $F_{a'}(C) = F_{b'}(C)$.

An *annotated* mGEN *network* is an mGEN network G' together with a valid function description F for G'.

A standard valid function description for a sound mGEN network is reachability, defined below.

Definition 3.19 (Reachability)**.** Fix an mGEN network G'. For $a' \in V'$, let $R_{a'}(C) :=$ TRUE if the extremal NO-instance $G(C)$ can reach a'.[17] The Boolean vector $R_{a'}$ is identified with the $\mathcal{C}$-vector $R_{a'}(C) := [\![C \text{ can reach } a']\!]$.

Note that $R_{s'} = \mathbf{1}$; and if G' is sound, then $R_{t'} = \mathbf{0}$.

Proposition 3.20 (Adjacent Reachability)**.** *If a' and b' are connected by an edge labeled l, where $l \in G(C)$, then $R_{a'}(C) = R_{b'}(C)$.*

Lemma 3.21 (Valid Function Description)**.** *An mGEN network G' has a valid function description if and only if G' is sound.*

Proof. If G' is sound, then the reachability vector assignment R is a valid function description for it. If G' is not sound, then G' accepts some NO-instance $G(C)$ for some $C \in \mathcal{C}$ (because $G(\mathcal{C})$ are all maxterms of GEN),[15] via some path P' connecting s' and t' labeled with literals in $G(C)$. If F is a function description for G' satisfying the adjacency condition (item (2)) in Definition 3.18, then for any two adjacent a' and b' on $V(P')$, $F_{a'}(C) = F_{b'}(C)$. Thus $F_{s'}(C) = F_{t'}(C)$ and F cannot be valid for G'. $\square$

We now introduce our key tool, invariant covers (Definition 3.23).

Definition 3.22 (l-Invariant)**.** For a literal l and an annotated mGEN network G' (with a valid function description F), a $\mathcal{C}$-vector g is *l-invariant on* G' if for any a' and b' in $V(G')$ connected by an edge labeled l, $\langle F_{a'}, g\rangle = \langle F_{b'}, g\rangle$. We say that g is *l-invariant* if g is l-invariant on G' for any annotated mGEN network G'.

Definition 3.23 (Invariant Cover)**.** Consider a YES instance P. A collection of $\mathcal{C}$-vectors $\{g_{P,\iota}\}_{\iota \in I}$ over some index set I forms an *invariant cover* (for P) if (1) for any positive literal $l \in G(P)$, there is an $\iota \in I$ so that $g_{P,\iota}$ is l-invariant; and (2) for any $\iota \in I$, we have $\langle \mathbf{1}, g_{P,\iota}\rangle = 1$.

The idea behind invariant covers is as follows. For each $\iota \in I$, item (2) of Definition 3.23 implies $\langle F_{s'}, g_{P,\iota}\rangle = 1$ and $\langle F_{t'}, g_{P,\iota}\rangle = 0$. Consider a path P' connecting s' and t', all of whose edges are labeled with literals from P. As we move from s' to t' along P', imagine that for each $\iota \in I$ there is a player trying to get $\langle F_{v'}, g_{P,\iota}\rangle$ from 1 to 0, where v' is the current node on P'. Whenever we move across an edge e' on P' with label $\lambda'(e') = l$, the players may all make progress towards this goal, except for the players

[17] Hence $R_{a'}$ is the truth table at node a' restricted to the extremal NO-instances $G(\mathcal{C})$.

$\iota \in I$ such that $g_{P,\iota}$ is l-invariant, who can make no progress. Since $l \in G(P)$, item (1) of Definition 3.23 ensures that at least one player cannot make progress. This implies that at some node b' on P', there will be a large discrepancy in progress among the players (Lemma 3.25), i. e., $\langle F_{b'}, g_{P,\iota_1} - g_{P,\iota_2}\rangle$ will be large for some $\iota_1, \iota_2 \in I$. Such a node b' is called a barrier node. If the $\mathcal{C}$-vectors $g_{P,\iota}$ are chosen carefully, this implies a large size lower bound (Lemma 3.29). This is made precise below (and the argument is phrased in terms of reachability R, the standard valid function description).

Lemma 3.24 (Barrier)**.** *Fix a* YES *instance P with an invariant cover $\{g_{P,\iota}\}_{\iota\in I}$ for P. Consider a path $P' =: \langle e'_1, e'_2, \dots, e'_\ell\rangle$ labeled with literals from P (that is, e'_j connects v'_{j-1} and v'_j and is labeled $\lambda'(e'_j) \in G(P)$) from $s' := v'_0$ to $t' := v'_\ell$. If $R_{s'} = \mathbf{1}$ and $R_{t'} = \mathbf{0}$, then there is a node b' on P', and distinct $\iota_1, \iota_2 \in I$, such that*

$$|\langle R_{b'}, g_{P,\iota_1} - g_{P,\iota_2}\rangle| \geq \frac{1}{\ell - 1}\,.$$

Proof. Apply Lemma 3.25 with $d_{j,\iota} := \langle R_{v'_j}, g_{P,\iota}\rangle$. □

Lemma 3.25 (Discrepancy in Progress)**.** *Consider real numbers $d_{j,\iota} \in \mathbb{R}$ for integer $0 \leq j \leq \ell$ and for index $\iota \in I$. If (1) for all $\iota \in I$, $d_{0,\iota} = 1$ and $d_{\ell,\iota} = 0$, and (2) for any $0 < j \leq \ell$, there is $\iota_j \in I$ such that $d_{j-1,\iota_j} = d_{j,\iota_j}$, then there exist j and distinct $\eta, \iota \in I$, such that*

$$|d_{j,\eta} - d_{j,\iota}| \geq \frac{1}{\ell - 1}\,.$$

Proof. For $0 \leq j \leq \ell$, let $m_j := \min_\iota d_{j,\iota}$ and $M_j := \max_\iota d_{j,\iota}$ and $s_j := M_j - m_j$ (s stands for *stretch* or *span*). Assumption (2) implies that $m_{j-1} \leq M_j = m_j + s_j$. Assumption (1) gives $m_\ell = s_\ell = 0$,

$$1 = m_0 \leq m_\ell + \sum_{0<j\leq\ell} s_j = \sum_{0<j<\ell} s_j\,,$$

and $s_j \geq \frac{1}{\ell-1}$ for some $0 < j < \ell$. □

We need a simple fact concerning the support of Fourier coefficients (Proposition 3.28), based on Definitions 3.26 and 3.27. The orthogonality in Fourier support gives the orthogonality in work done to different pyramids.

Definition 3.26 (Fourier Support)**.** For a pyramid P, say a $\mathcal{C}$-vector g is *P-supported* if

1. g depends only on coloring in P, i. e., $g(C_1) = g(C_2)$ if $v \in C_1 \Leftrightarrow v \in C_2$ for all $v \in V(P)$; or *equivalently*
2. $\hat{g}(U) \neq 0$ only when $U \subseteq V(P)$.

Definition 3.27 (High Frequency Support)**.** We introduce *cut-off degree* and *agreement degree* for describing the Fourier support.

1. A $\mathcal{C}$-vector g has *cut-off degree* k if $\hat{g}(U) \neq 0$ only when $|U| \geq k$.
2. A collection of $\mathcal{C}$-vectors $\{g_\iota\}_{\iota\in I}$ *agrees up to degree* k if for all $U \in \mathcal{C}$ where $|U| \leq k$, we have $\hat{g}_{\iota_1}(U) = \hat{g}_{\iota_2}(U)$ for $\iota_1, \iota_2 \in I$.

Proposition 3.28 (Cut-Off and Agreement)**.** *For a collection of $\mathcal{C}$-vectors $\{g_\iota\}_{\iota\in I}$, the following are equivalent:*

1. *for any $\iota_1,\iota_2 \in I$, the difference $g_{\iota_1} - g_{\iota_2}$ has cut-off degree $k+1$; and*
2. *$\{g_\iota\}_{\iota\in I}$ agrees up to degree k.*

Later, we will construct an invariant cover agreeing up to degree $k = \Omega(h)$ and of "length" $\sigma \le O(k\log n)$ (with a small enough constant in $O(\cdot)$), giving a size lower bound of $2^{\Omega(h\log n)} = n^{\Omega(h)}$ via Lemma 3.29.

Lemma 3.29 (Size Lower Bound from Invariant Cover)**.** *For $h \le n^{1/8}$, if there is an invariant cover $\{g_{P,\iota}\}_{\iota\in I}$ for a pyramid* YES*-instance P of height h such that (1) $g_{P,\iota}$ is P-supported, (2) $\{g_{P,\iota}\}_{\iota\in I}$ agrees up to degree $k \le \binom{h+1}{2}$, and (3) $\|g_{P,\iota}\| \le 2^\sigma$ for $\iota \in I$; then any sound m*GEN *network G' having n' nodes for* GEN *of size n and accepting all pyramid* YES*-instances, satisfies*

$$n' \ge 2^{(k\log n)/40-(\sigma+1)/2}.$$

Proof. Fix a pyramid P of height h. Its YES instance $G(P)$ is accepted by G' via some path P' labeled with literals from $G(P)$. Lemma 3.24 gives a node b'_P and a $\mathcal{C}$-vector $g_P := g_{P,\iota_1} - g_{P,\iota_2}$ such that $|\langle R_{b'_P}, g_P\rangle| \ge 1/n'$. Note that $\|g_P\| \le \|g_{P,\iota_1}\| + \|g_{P,\iota_2}\| \le 2^{\sigma+1}$. Also, g_P has cut-off degree $k+1$ by Proposition 3.28, and is P-supported.

Let $q := n^{k/10}$, $m := \binom{h+1}{2}$, then Lemma 2.2 gives q sets $Q_1, Q_2, \dots, Q_q \subseteq [N]$ satisfying $|Q_i| = m$ and $|Q_i \cap Q_j| \le k$ for $1 \le i \ne j \le q$, where

$$N = \exp\left(\frac{k\ln n}{10k} + 1\right)\frac{m^2}{k} \le en^{1/10}m^2 \ll n\,.$$

For each Q_i, construct a pyramid P_i by identifying Q_i arbitrarily with a pyramid graph of height h. Now the previous paragraph gives $b'_i := b'_{P_i}$ and $g_i := g_{P_i}$. Normalize $\tilde{g}_i := g_i/\|g_i\|$, then $\{\tilde{g}_i\}_{1\le i\le q}$ is orthonormal (having disjoint Fourier support). Note that $1 \ge \|R_{b'}\|^2 \ge \sum_i |\langle R_{b'}, \tilde{g}_i\rangle|^2$ for any $b' \in V'$. Now

$$n' = \sum_{a'\in V'} 1 \ge \sum_i \sum_{a'\in V'} |\langle R_{a'}, \tilde{g}_i\rangle|^2 \ge \sum_i |\langle R_{b'_i}, \tilde{g}_i\rangle|^2 \ge q\left(\frac{1}{n'2^{\sigma+1}}\right)^2,$$

since

$$|\langle R_{b'_i}, \tilde{g}_i\rangle| \ge \frac{1}{n'}\frac{1}{\|g_i\|} \ge \frac{1}{n'2^{\sigma+1}}\,.$$

It follows that $(n')^3 \ge q/2^{2\sigma+2} = n^{k/10}/2^{2\sigma+2}$. □

3.3.3 Reduction to reversible pebbling

This subsection reuses the lower bound for reversible pebbling mGEN networks (Lemma 3.11) for constructing invariant covers. Recall that each $\mathcal{C}$-vector $g_{P,\iota}$ in an invariant cover is l-invariant for some literal (triple) l. Note that being l-invariant is a strong condition: $g_{P,\iota}$ must satisfy a linear algebraic relation with respect to *any* mGEN network. However, it turns out that l-invariant vectors can be constructed using a *single* mGEN network (Definition 3.31 and Corollary 3.36), which is even a reversible pebbling mGEN network. Indeed, G'_P (Definition 3.31) is suitable for this task, because "any reversible pebbling strategy to pebble P can be lifted to, or seen as a subgraph (i. e., edge subset) of, G'_P with a matching function description K." Hence the name *universal*.

Definition 3.30 (ζ-Function[18]). Identify a reversible pebbling configuration $K_{a'} \in \mathcal{K}$ with the $\mathcal{C}$-vector $K_{a'}(C) := [\![K_{a'} \subseteq C]\!]$ for $C \in \mathcal{C}$.[16]

Definition 3.31 (Universal Reversible Pebbling Switching Network). Given a pyramid P, consider $G'_P := (V'_P, E'_P, s'_P, t'_P, \lambda'_P)$ where $V'_P := \mathsf{PowerSet}\big(V(P)\big) \cup \{K_{t'}\} \subseteq \mathcal{K}$ is the collection of all proper reversible pebbling configurations over elements of P, plus the improper reversible pebbling configuration $K_{t'} := [n]$. Hence each $a' \in V'_P$ is identified and naturally associated with its reversible pebbling configuration (i. e., $K_{a'} := a' \cup \{s\} \in \mathcal{K}$ as in Definition 3.6), with the obvious distinguished nodes s'_P and t'_P so that $K_{s'_P} = K_{s'}$ and $K_{t'_P} = K_{t'}$. Now a' and b' in V'_P are connected by an edge e' in E'_P with label a triple $\lambda'_P(e') = l \in G(P)$ iff $K_{a'} + l = K_{b'} + l$. Then G'_P is a reversible pebbling mGEN network annotated with a valid function description $K \colon V'_P \to \mathbb{R}^{\mathcal{C}}$ (Definition 3.18 and Proposition 3.15).

To construct invariants (Definition 3.22) using the universal network (Definition 3.31), we need an alternative, equivalent characterization of invariants (Lemma 3.33) concerning the spatial[19] support of $\mathcal{C}$-vectors (Definition 3.32).

Definition 3.32 (l-Definite). For a literal l, say that a $\mathcal{C}$-vector g is *l-definite* if $g(C) \neq 0$ implies $l \in G(C)$ (Definition 3.14).

Lemma 3.33 (Invariant $\Leftrightarrow$ Definite). *g is l-invariant iff g is l-definite.*

Proof. For the $\Leftarrow$ direction, on any annotated mGEN network G' with a valid function description F, when a' and b' are connected by an edge labeled l, $F_{a'}(C)g(C) = F_{b'}(C)g(C)$ for any C, because either (i) $g(C) = 0$, or (ii) $l \in G(C)$, hence $F_{a'}(C) = F_{b'}(C)$ (item 2 of Definition 3.18). The $\Rightarrow$ direction follows from Lemma 3.34. $\square$

The following proof of Lemma 3.34, i. e., backward induction and Lemma 3.35, was communicated to us by Yuval Filmus and Robert Robere, substantially simplifying proofs and presentations in earlier versions.

Lemma 3.34 (Reversible Pebbling Invariant). *If g is l-invariant on G'_P, then g is l-definite.*

Proof. If $l \notin G(C)$ and g is l-invariant on G'_P, then $g(C) = 0$ by a backward induction on C as follows. Note that $v \notin C$ if $l =: w \wedge u \to v \notin G(C)$, and

$$g(C) = \sum_{C \subseteq D \subseteq \tilde{V} \setminus \{v\}} g(D) - \sum_{C \subset D \subseteq \tilde{V} \setminus \{v\}} g(D)\,.$$

The first term is zero by Lemma 3.35, and the second term is also zero if (1) $C = \tilde{V} \setminus \{v\}$ vacuously, or (2) $g(D) = 0$ for all $C \subset D \subseteq \tilde{V} \setminus \{v\}$ by induction hypothesis. $\square$

[18]The $\mathcal{C}$-vector $K_{a'}(\cdot)$ can be identified with $\zeta(a', \cdot)$ of the incidence algebra over the partially ordered set $\mathcal{C}$ of proper reversible pebbling configurations. For a fixed pyramid P, the corresponding inverse $g^{a'}(\cdot)$ (see Proposition 3.40) can basically be identified with $\mu(\cdot, a')$, interpreting the inner product $\langle \cdot, \cdot \rangle$ (Definition 3.16) as a convolution (Proposition 3.40).

[19]Spatial as in spatial domain, as the dual to frequency domain.

Lemma 3.35 (Inclusion-Exclusion Principle)**.** *If g is l-invariant on G'_P, and $l =: w \wedge u \to v \notin G(C)$, then* $\sum_{C \subseteq D \subseteq \tilde{V} \setminus \{v\}} g(D) = 0$.

Proof. Let $a' := C$ and $b' := C \cup \{v\}$, then a' and b' are connected by an edge labeled l on G'_P, for $C + l = C \cup \{v\}$. Since g is l-invariant on G'_P, $\langle K_{a'}, g\rangle = \langle K_{b'}, g\rangle$. Hence

$$0 = |\mathcal{C}|\langle g, K_{a'} - K_{b'}\rangle = \sum_D g(D)\big(\llbracket a' \subseteq D\rrbracket - \llbracket b' \subseteq D\rrbracket\big) = \sum_{C \subseteq D \subseteq \tilde{V} \setminus \{v\}} g(D)\,. \qquad \square$$

Corollary 3.36 (Reversible Pebbling Invariant)**.** *g is l-invariant iff g is l-invariant on G'_P.*

Proof. The $\Leftarrow$ direction follows from Lemma 3.34 and the $\Leftarrow$ direction of Lemma 3.33. $\square$

3.3.4 Invariant cover from the universal network

For the many properties required in the lower bound framework (Lemma 3.29) concerning Fourier support, Corollary 3.36 intuitively handles the condition of l-invariance (implicit in an invariant cover). Other properties are easier to satisfy than l-invariance, except perhaps the property of agreement, which is handled by the notion of l-accessible reversible pebbling configurations (Definition 3.38) relative to a barrier (Definition 3.37). Roughly, different l-accessible reversible pebbling configurations A'_l (for different literals l) agree with each other over the common accessible reversible pebbling configurations A', which translates to the agreement condition (see Lemma 3.41).

Definition 3.37 (Barrier)**.** Consider a set of nodes $B' \subseteq V'_P$. A path P' on G'_P *intersects* B' if $V(P') \cap B' \neq \{\}$. A collection of reversible pebbling configurations $B' \subset V'_P$ is a *barrier* if $s'_P \notin B'$ and $t'_P \notin B'$ and every path P' from s'_P to t'_P on G'_P intersects B'.

Definition 3.38 (l-Accessible Reversible Pebbling Configurations)**.** Relative to a barrier $B' \subset V'_P$, we define the collection of *accessible* reversible pebbling configurations

$$A' := \left\{a' \in V'_P \colon \exists \text{ a path } P' \text{ connecting } s'_P \text{ and } a' \text{ and } P' \text{ does } \textit{not} \text{ intersect } B'\right\}.$$

We want an extended notion of l-accessible reversible pebbling configurations $A'_l \supseteq A'$ where we allow intersection caused by edges labeled l, defined by

$$A'_l := \left\{a' \in V'_P \colon a' \in A' \text{ or } a' \text{ is adjacent to } b' \in A' \text{ by an edge labeled } l\right\}.$$

That is, all edges crossing A'_l are *not* labeled with l. Indeed, if a' is adjacent to $b' \in A'$ by an edge labeled $l =: w \wedge u \to v$, then a' and b' differ by a reversible pebbling move (Remark 3.3) to pebble or unpebble v, since t is not involved if $b' \in A'$. It follows that if a' and b' are connected by an edge labeled $l =: w \wedge u \to v$, and $a' \in A'_l$, then $b' \in A'_l$ as well (they differ at most by a reversible pebbling move (Remark 3.3) to pebble or unpebble v).

Remark 3.39 (Möbius Inversion)**.** To construct an invariant cover $\{g_{P,l}\}_{l \in G(P)}$ (Lemma 3.41) where each $g_{P,l}$ is P-supported (for use in Lemma 3.29), it suffices to restrict attention to P-supported vectors, which form a subspace of $\mathcal{C}$-vectors. Note that the $\mathcal{C}$-vector $K_{a'}$ is P-supported iff $a' \subseteq V(P) = Q$

(Definition 3.30), i. e., if $K_{a'}$ is over elements of P. Since $K_{a'}(\cdot) = \zeta(a',\cdot)$, the collection $\{K_{a'}\}_{a'\subseteq V(P)}$ can be identified with the ζ-function of (the incidence algebra over) the partially ordered set $\mathsf{PowerSet}\big(V(P)\big)$ of proper reversible pebbling configurations over elements of P. Hence $\{K_{a'}\}_{a'\subseteq V(P)}$ forms a basis for the subspace of P-supported $\mathcal{C}$-vectors, admitting a dual basis $\{g^{b'}\}_{b'\in V(P)}$ satisfying $\langle K_{a'}, g^{b'}\rangle = [\![a' = b']\!]$ (Proposition 3.40),[16] given by an inclusion-exclusion over $\mathsf{PowerSet}\big(V(P)\big)$ (i. e., a Möbius inversion). Hence if $g_{P,l} := \sum_{a'\in A'_l} g^{a'}$, then $\langle K_{b'}, g_{P,l}\rangle = [\![b' \in A'_l]\!]$, indicating whether $b' \in A'_l$. Thus for any a' and b' on G'_P connected by an edge labeled l, it follows that $\langle K_{a'}, g_{P,l}\rangle = [\![a' \in A'_l]\!] = [\![b' \in A'_l]\!] = \langle K_{b'}, g_{P,l}\rangle$ by the construction of A'_l (Definition 3.38). So $g_{P,l}$ is l-invariant on G'_P, hence l-invariant (Corollary 3.36).

Proposition 3.40 (Dual Vector to $K_{a'}$)**.** *Fix a subset $Q = V(P) \subseteq \tilde{V} \subset [n]$ of size $|Q|$. We write $\bar{C} := C \cap Q$ for $C \in \mathcal{C}$. For any subset $a' \subseteq Q$, define a P-supported $\mathcal{C}$-vector $g^{a'}$ by*

$$g^{a'}(C) := 2^{|Q|}[\![\bar{C} \subseteq a']\!](-1)^{|a'|-|\bar{C}|}\,.$$

Then (1) $\langle K_{b'}, g^{a'}\rangle = [\![a' = b']\!]$ for $a', b' \subseteq Q$;[16] *and (2) $\hat{g}^{a'}(U) = (-2)^{|a'|}[\![a' \subseteq U]\!]$. In particular, $\hat{g}^{a'}$ is supported on $U \supseteq a'$, i. e., $\hat{g}^{a'}(U) \neq 0$ only when $U \supseteq a'$.*

Proof. Note that $g^{a'}$ is P-supported by definition, hence

$$\langle K_{b'}, g^{a'}\rangle = \frac{1}{|\mathcal{C}|}\sum_{C\in\mathcal{C}} K_{b'}(C)g^{a'}(C) = \frac{1}{2^{|Q|}}\sum_{\bar{C}\subseteq Q} K_{b'}(\bar{C})g^{a'}(\bar{C}) = \sum_{b'\subseteq\bar{C}\subseteq a'}(-1)^{|a'|-|\bar{C}|} = [\![a' = b']\!]\,,$$

on observing that $(-1)^{|a'|-|\bar{C}|} = \mu(\bar{C}, a')$ is the μ-function of (the incidence algebra over) the partially ordered set $\mathsf{PowerSet}\big(V(P)\big)$. Thus (1) holds. For (2), since $g^{a'}$ is P-supported, focus on $U \subseteq Q$, then

$$\hat{g}^{a'}(U) = \langle \chi_U, g^{a'}\rangle = \sum_{\bar{C}\subseteq a'}(-1)^{|\bar{C}\cap U|+|a'|-|\bar{C}|}\,,$$

which is zero (due to cancellation) unless $a' \subseteq U$, in which case it gives $\sum_{\bar{C}\subseteq a'}(-1)^{|a'|} = (-2)^{|a'|}$. $\square$

Lemma 3.41 (Constructing Invariants)**.** *For any pyramid P of height $h \geq 2$, let $m := \binom{h+1}{2}$, for any positive literal $l \in G(P)$, there is a $\mathcal{C}$-vector $g_{P,l}$ such that (1) $g_{P,l}$ is P-supported, (2) $g_{P,l}$ is l-invariant, (3) $\langle \mathbf{1}, g_{P,l}\rangle = 1$, (4) for all $U \subseteq \tilde{V}$ where $|U| \leq k := h-1$, we have $\hat{g}_{P,l}(U)$ is independent of l,*[20] *(5) for all $U \subseteq \tilde{V}$, $|\hat{g}_{P,l}(U)| \leq 2^{\tau}$ where $\tau := 2h\log(em/h)$, and (6) $\|g_{P,l}\| \leq 2^{\sigma}$ where $\sigma := m/2 + \tau$.*

Proof. Let $B' := \big\{a' \in V'_P \setminus \{t'_P\} \colon |K_{a'} \setminus \{s\}| = h\big\}$ be the collection of reversible pebbling configurations (on the universal reversible pebbling switching network G'_P) having exactly h vertices on $V(P)$, which is a barrier by Lemma 3.11. Let A'_l be the l-accessible reversible pebbling configurations relative to B'. Note that $t'_P \notin A'_l$ since $h \ll \binom{h+1}{2}$. Now let $g_{P,l} := \sum_{a'\in A'_l} g^{a'}$, where $g^{a'}$ is the *dual vector* to $K_{a'}$ (Proposition 3.40). Each $g^{a'}$ is P-supported, hence (1) holds. The end of Remark 3.39 establishes (2). For (3), note that $\mathbf{1} = \chi_{\{\}}$ and $\hat{g}^{a'}(\{\}) \neq 0$ only when $a' = s'_P$, and $\hat{g}^{s'_P}(\{\}) = 1$ (Proposition 3.40). For (4), note that if a' is such that $|K_{a'} \setminus \{s\}| < h$, then $a' \in A'_l$ iff $a' \in A'$, hence $[\![a' \in A'_l]\!] = [\![a' \in A']\!]$ is independent

[20] A collection $\big\{g_{P,l}(U)\big\}_l$ is independent of l if $g_{P,l_1}(U) = g_{P,l_2}(U)$ for any l_1, l_2.

of l for such a'. Also note that $\hat{g}^{a'}(U) \neq 0$ only when $a' \subseteq U$ by Proposition 3.40. For (5), note that if $a' \in A'_l$, then $|K_{a'} \setminus \{s\}| \leq h$. Also recall

$$\binom{m}{\leq h} := \sum_{i=0}^{h} \binom{m}{i} \leq \left(\frac{\mathrm{e}m}{h}\right)^h .$$

Since $\hat{g}_{P,l}(U) = \sum_{a' \in A'_l} \hat{g}^{a'}(U)$,

$$\left|\hat{g}_{P,l}(U)\right| \leq \sum_{a' \in A'_l} \left|\hat{g}^{a'}(U)\right| \leq \sum_{a' \in A'_l} 2^{|a'|} \leq \binom{m}{\leq h} 2^h \leq \left(\frac{\mathrm{e}m}{h}\right)^h 2^h = 2^{h\log(\mathrm{e}m/h)+h} .$$

using Proposition 3.40. For (6), $\langle g_{P,l}, g_{P,l}\rangle = \sum_{U \subseteq Q} \hat{g}_{P,l}(U)^2 \leq 2^m 2^{2\tau}$. □

3.3.5 Tight size bound

Note that the lower bound (Lemma 3.29) gets stronger as the invariant cover gets shorter (having a smaller σ). And for tight size bounds (Theorems 3.46 and 3.48), the first term $\Theta(k \log n)$ has to dominate the second term $\Theta(\sigma)$. To this end, we need yet another alternative, equivalent characterization of invariants (Definition 3.22) concerning Fourier support (Lemma 3.44). This characterization allows us to chop off high frequency Fourier coefficients from $g_{P,l}$ to shorten the vectors (Lemma 3.45). Recall that we focus on literals $l = w \wedge u \to v$ where $w \neq v$ and $u \neq v$. Further, assume $w \neq t$ and $u \neq t$ and $v \neq s$, which is the case for our analysis (Lemma 3.29) and would simplify the treatment in Lemma 3.44.

Propositions 3.42 and 3.43 are simple facts about representing Boolean logic in Fourier analysis.

Proposition 3.42 (Arithmetization)**.** *Fix $v \in [n]$. Define the $\mathcal{C}$-vectors α_v and β_v by*

$$\alpha_v(C) := \begin{cases} \frac{1}{2}\big(\chi_{\{\}}(C) - \chi_{\{v\}}(C)\big) & \text{if } v \in \tilde{V}, \\ \chi_{\{\}}(C) & \text{if } v = s, \\ 0 & \text{if } v = t. \end{cases} \quad \text{and} \quad \beta_v(C) := \begin{cases} \frac{1}{2}\big(\chi_{\{\}}(C) + \chi_{\{v\}}(C)\big) & \text{if } v \in \tilde{V}, \\ 0 & \text{if } v = s, \\ \chi_{\{\}}(C) & \text{if } v = t. \end{cases}$$

Then for any $C \in \mathcal{C}$, we have $[\![v \in C \cup \{s\}]\!] = \alpha_v(C)$ and $[\![v \notin C \cup \{s\}]\!] = \beta_v(C)$.

Proposition 3.43 (Orbital Average)**.** *For $Z \subseteq \tilde{V}$ and $v \in \tilde{V} \setminus Z$, define $Z_v := Z \cup \{v\}$, then*

$$\big(\chi_{\{\}} - \chi_{\{v\}}\big)\big(\hat{g}(Z)\chi_Z + \hat{g}(Z_v)\chi_{Z_v}\big) = \big(\hat{g}(Z) - \hat{g}(Z_v)\big)\big(\chi_Z - \chi_{Z_v}\big)$$

and

$$\big(\chi_{\{\}} + \chi_{\{v\}}\big)\big(\hat{g}(Z)\chi_Z + \hat{g}(Z_v)\chi_{Z_v}\big) = \big(\hat{g}(Z) + \hat{g}(Z_v)\big)\big(\chi_Z + \chi_{Z_v}\big) .$$

Proof. When $v \notin Z$, we have $\chi_{Z_v}(C) = \chi_Z(C)\chi_{\{v\}}(C)$, so (under co-ordinate-wise multiplication)

$$\big(\chi_{\{\}} - \chi_{\{v\}}\big)\chi_Z = \chi_Z - \chi_{Z_v} = \big(\chi_{\{v\}} - \chi_{\{\}}\big)\chi_{Z_v} ,$$

giving the first equality. The second equality is proved similarly. □

Lemma 3.44 (Fourier Invariant)**.** *Fix a triple $l := w \wedge u \to v$. Let $\psi(l) := \{w,u,v\} \cap \tilde{V}$. A $\mathcal{C}$-vector g is l-invariant iff for any $Z \subseteq \tilde{V} \setminus \psi(l)$, we have*

$$0 = \xi_{g,l}(Z) := \sum_{X \subseteq \psi(l)} \chi_{\tilde{V} \cap \{w,u\}}(X)\hat{g}(Z \cup X)\,.$$

Proof. g is l-invariant iff g is l-definite (Lemma 3.33) iff $g(C) = [\![l \in G(C)]\!] g(C)$ for all $C \in \mathcal{C}$ (Definition 3.32). If $l = w \wedge u \to v$ for $w,u,v \in [n]$, then $[\![l \in G(C)]\!] = 1 - [\![w \in C_s]\!][\![u \in C_s]\!][\![v \notin C_s]\!]$ where $C_s := C \cup \{s\}$ (Definition 3.14). Therefore, for any $C \in \mathcal{C}$, we have

$$g(C) = [\![l \in G(C)]\!] g(C) \qquad \text{iff} \qquad [\![w \in C_s]\!][\![u \in C_s]\!][\![v \notin C_s]\!] g(C) = 0\,.$$

Succinctly, g is l-invariant iff (under co-ordinate-wise multiplication)

$$[\![w \in \cdot \cup \{s\}]\!][\![u \in \cdot \cup \{s\}]\!][\![v \notin \cdot \cup \{s\}]\!] g = \mathbf{0}$$

as $\mathcal{C}$-vectors, iff (by Proposition 3.42) $\alpha_w \alpha_u \beta_v g = \mathbf{0}$ as $\mathcal{C}$-vectors. Fourier expand g as

$$g = \sum_{U \subseteq \tilde{V}} \hat{g}(U)\chi_U = \sum_{Z \subseteq \tilde{V} \setminus \psi(l)} \sum_{Y \subseteq \psi(l)} \hat{g}(Z \cup Y)\chi_{Z \cup Y}\,. \tag{3.1}$$

Now g is l-invariant iff (under co-ordinate-wise multiplication as $\mathcal{C}$-vectors)

$$\mathbf{0} = \alpha_w \alpha_u \beta_v \sum_{Z \subseteq \tilde{V} \setminus \psi(l)} \sum_{Y \subseteq \psi(l)} \hat{g}(Z \cup Y)\chi_{Z \cup Y}\,,$$

which (by Proposition 3.43, recall $w \neq t$, $u \neq t$, $v \neq s$) is equivalent to

$$\begin{aligned}
\mathbf{0} &= \sum_{Z \subseteq \tilde{V} \setminus \psi(l)} \Big(\sum_{X \subseteq \psi(l)} \chi_{\tilde{V} \cap \{w,u\}}(X)\hat{g}(Z \cup X) \sum_{Y \subseteq \psi(l)} \chi_{\tilde{V} \cap \{w,u\}}(Y)\chi_{Z \cup Y} \Big) \\
&= \sum_{Z \subseteq \tilde{V} \setminus \psi(l)} \xi_{g,l}(Z)\eta(Z)\,,
\end{aligned}$$

where $\eta(Z) := \sum_{Y \subseteq \psi(l)} \chi_{\tilde{V} \cap \{w,u\}}(Y)\chi_{Z \cup Y}$. Note that $\{\eta(Z)\}_Z$ are orthogonal and non-zero, hence linearly independent. ☐

Lemma 3.45 (Low-Pass Invariant)**.** *Fix natural numbers k and τ. **If** for a pyramid P, for any positive literal $l \in G(P)$, there is a $\mathcal{C}$-vector $g_{P,l}$ such that (1) $g_{P,l}$ is P-supported, (2) $g_{P,l}$ is l-invariant, (3) $\langle \mathbf{1}, g_{P,l} \rangle = 1$, (4) for all $U \subseteq \tilde{V}$ where $|U| \leq k$, we have $\hat{g}_{P,l}(U)$ is independent of l,*[20] *and (5) for all $U \subseteq \tilde{V}$, $|\hat{g}_{P,l}(U)| \leq 2^{\tau}$; **then** for any positive literal $l \in G(P)$, there is a $\mathcal{C}$-vector $\tilde{g}_{P,l}$ satisfying (1), (2), (3), (4), and (5), plus (6) $\|\tilde{g}_{P,l}\| \leq 2^{\sigma}$ where $\sigma := \big((k+4)\log m\big) + \tau$.*

Proof. For $l \in G(P)$, expand $g_{P,l} = \sum_{C \in \mathcal{C}} \hat{g}_{P,l}(C)\chi_C$ as a $\mathcal{C}$-vector. If $l =: w \wedge u \to v$, define

$$\mathcal{C}_{l,k} := \big\{C \in \mathcal{C} \colon |C \setminus \psi(l)| \leq k\big\} \subseteq \mathcal{C}$$

for $\psi(l) := \tilde{V} \cap \{w,u,v\}$, and $\tilde{g}_{P,l} := \sum_{C \in \mathcal{C}_{l,k}} \hat{g}_{P,l}(C)\chi_C$ as a $\mathcal{C}$-vector by chopping off the high frequency Fourier coefficients from $g_{P,l}$. Note that $\langle \tilde{g}_{P,l}, \chi_U \rangle \neq 0$ implies $\langle g_{P,l}, \chi_U \rangle \neq 0$ for any $U \in \mathcal{C}$. Now $\tilde{g}_{P,l}$

is P-supported due to the second item of Definition 3.26, hence (1) holds. Since $g_{P,l}$ is l-invariant, by Lemma 3.44, $\xi_{g,l}(Z) = 0$ for any $Z \subseteq \tilde{V} \setminus \psi(l)$. It follows that $\xi_{\tilde{g},l}(Z) = 0$ for any such Z, as either (i) for some $X \subseteq \psi(l)$, $Z \cup X \in \mathcal{C}_{l,k}$, hence $Z \cup X \in \mathcal{C}_{l,k}$ for all such X, thus $\xi_{\tilde{g},l}(Z) = \xi_{g,l}(Z)$; or (ii) for all $X \subseteq \psi(l)$, $Z \cup X \notin \mathcal{C}_{l,k}$, thus $\xi_{\tilde{g},l}(Z) = 0$. So $\tilde{g}_{P,l}$ is l-invariant by Lemma 3.44 and (2) holds. For (3), $\mathbf{1} = \chi_{\{\}}$ and $\langle \tilde{g}_{P,l}, \chi_{\{\}} \rangle = \langle g_{P,l}, \chi_{\{\}} \rangle$. For (4), when $U \subseteq \tilde{V}$ has $|U| \le k$, $\langle \tilde{g}_{P,l}, \chi_U \rangle = \langle g_{P,l}, \chi_U \rangle$. (5) follows since $|\langle \tilde{g}_{P,l}, \chi_U \rangle| \le |\langle g_{P,l}, \chi_U \rangle|$ for all $U \subseteq \tilde{V}$. Note that $|C| \le k+3$ for $C \in \mathcal{C}_{l,k}$, hence

$$\langle \tilde{g}_{P,l}, \tilde{g}_{P,l} \rangle = \sum_{C \in \mathcal{C}_{l,k}} \hat{g}^2_{P,l}(C) \le \binom{m}{\le k+3} 2^{2\tau} \le m^{2k+6} 2^{2\tau} . \qquad \square$$

Theorem 3.46 (Size Lower Bound)**.** *For any $4 \le h \le n^{1/600}$, any sound m*GEN *network G' having n' nodes for* GEN *of size n, and accepting all pyramid* YES*-instances of height h, satisfies*

$$n' \ge 2^{(h \log n)/400 - 1/2} = n^{\Theta(h)} .$$

Proof. By Lemma 3.41, boosted by Lemma 3.45, there is an invariant cover $\{g_{P,l}\}_{l \in G(P)}$ for P such that (1) $g_{P,l}$ is P-supported and (2) $\{g_{P,l}\}_{l \in G(P)}$ agrees up to degree $k := h-1$, and (3) $\|g_{P,l}\| \le 2^{\sigma}$ with $\sigma \le (k+4)\log m + 4h \log m \le 6h \log m$. Now Lemma 3.29 gives

$$\log n' \ge \frac{k}{40} \log n - \frac{1}{2}\sigma - \frac{1}{2} \ge \frac{h}{80} \log n - 3h \log m - \frac{1}{2} \ge \frac{h}{400} \log n - \frac{1}{2} . \qquad \square$$

Theorem 3.46 gives a size lower bound for mGEN networks solving the generation problem, by analyzing pyramid YES-instances. However, the general generation problem may be harder and require larger mGEN networks. To get a tight bound (and thus defending the title of this paper), we construct below a monotone (decision) problem computed by a (uniform) mGEN network (Definition 3.47), allowing us to reuse Theorem 3.46 for a tight size bound in Theorem 3.48.

Definition 3.47 (Universal Degree-h Reversible Pebbling Switching Network)**.** For a size parameter n and $h \le n$, consider

$$G'_{n,h} := (V'_h, E'_h, s'_h, t'_h, \lambda'_h)$$

where

$$V'_h := \binom{\tilde{V}}{\le h} \cup K_{t'} := \big\{ V \subseteq [n] \setminus \{s,t\} \colon |V| \le h \big\} \cup \{K_{t'}\}$$

is the collection of proper reversible pebbling configurations of size at most h (excluding s), plus the improper reversible pebbling configuration. Hence each $a' \in V'_h$ is identified and naturally associated with its reversible pebbling configuration ($K_{a'} := a' \cup \{s\} \in \mathcal{K}$ as in Definition 3.6), with the obvious distinguished nodes s'_h and t'_h so that $K_{s'_h} = K_{s'}$ and $K_{t'_h} = K_{t'}$. Now a' and b' in V'_h are connected by an edge e' in E'_h labeled with a triple $\lambda'_h(e') = l$ iff $K_{a'} + l = K_{b'} + l$. Then $G'_{n,h}$ is a reversible pebbling mGEN network annotated with a valid function description $K \colon V'_P \to \mathbb{R}^{\mathcal{C}}$ (Definition 3.18 and Proposition 3.15).

Recall that $f_{G'_{n,h}}$ is the function computed by $G'_{n,h}$, which is monotone. Note that $G'_{n,h}$ has

$$\binom{n-2}{\le h} + 1 = n^{O(h)}$$

nodes (and hence $n^{O(h)}$ edges).

Theorem 3.48 (Tight Size Bound)**.** *For any $12 \le h \le 3n^{1/600}$, any m*GEN *network G' having n' nodes and computing $f_{G'_{n,h}}$ satisfies*

$$n' \ge \frac{1}{2} n^{h/1200} = n^{\Theta(h)} .$$

Proof. Note that $G'_{n,h}$ is sound and accepts all pyramid YES-instances of height $h/3$ (by simulating a reversible pebbling strategy), and so is any G' computing $f_{G'_{n,h}}$. Now apply Theorem 3.46. □

4 Lower bound for cliques

This section proves the lower bound for cliques as Theorem 4.19. After defining the problem and the model below (Definitions 4.1 and 4.2), Section 4.1 introduces the Fourier analytic framework over extremal NO-instances, Section 4.2 describes the lower bound framework assuming the existence of an invariant cover, and Section 4.3 constructs the invariant cover. These subsections clearly share many similarities with the lower bound proof in Section 3.

Definition 4.1 (Clique Problem)**.** For a size parameter n, let $V := [n]$ be n vertices. For $2 \le k \le n$, the *k-clique problem* (k-CLIQUE) receives as input a subset $E \subseteq \binom{V}{2}$ of edges. k-CLIQUE accepts input E if the graph (V,E) has a k-clique, i. e., if there is $U \subseteq V$, $|U| = k$, and $\{u,v\} \in E$ for all $u \neq v \in U$. The value of E is represented as $\binom{n}{2}$ Boolean variables.

Definition 4.2 (Monotone Switching Networks for CLIQUE)**.** We say that a switching network is a monotone switching network for CLIQUE (mCLIQUE network), if each edge $e' \in E'$ is labeled with $\lambda'(e') = \{u,v\}$ for some $u \neq v \in V$. Given an instance for k-CLIQUE with data E, the literal $\lambda'(e') = \{u,v\}$ is *in the instance* if $\{u,v\} \in E$.

4.1 Fourier analysis on extremal instances

The analysis focuses on the YES-instances of CLIQUE having a structure of a k-clique, and the NO-instances that are $(k-1)$-colorable.[21] In particular, those YES-instances are minterms and those NO-instances are 'similar' to maxterms of CLIQUE as a monotone Boolean function, and they are respectively "hardest to accept" and "hardest to reject." Fourier analysis will be done on the truth tables restricted to the $(k-1)$-colorable instances (Definition 4.7). These definitions for CLIQUE are analogous to the ones for GEN in Section 3.3.1.

Definition 4.3 (k-clique YES-Instances)**.** An instance $G(P) \in \{0,1\}^{\binom{n}{2}}$ of CLIQUE forms a *k-clique* YES-*instance* if there exists $Q \subseteq V$, $|Q| = k$ such that $\{u,v\} \in E$ iff $u \neq v$ and $u,v \in Q$. In this case, the YES-instance is simply called a k-clique (also denoted by P), and $V(P) := Q \subseteq V$.

Definition 4.4 (Extremal NO-Instances)**.** Let $\mathcal{C} := \{C\colon V \to [k-1]\} \cong [k-1]^V$ be the collection of $k-1$ colorings of V. A coloring $C \in \mathcal{C}$ is associated with the extremal NO-instance $G(C)$ where an edge $\{u,v\} \notin G(C)$ iff $C(u) = C(v)$ for $u,v \in V$.

[21] This consideration is standard in the study of the monotone complexity of k-CLIQUE (e. g., [46,55]), and similar ideas date back at least to Razborov [48].

Definition 4.5 (Inner Product Space of Colorings)**.** A $\mathcal{C}$-vector is a complex vector with index set $\mathcal{C}$, equivalently a function from $\mathcal{C}$ to $\mathbb{C}$. For two $\mathcal{C}$-vectors f and g, we define

$$\langle f,g\rangle := \frac{1}{|\mathcal{C}|}\sum_{C\in\mathcal{C}} f(C)\overline{g(C)}$$

as their inner product, inducing the norm $\|f\| := \langle f,f\rangle^{1/2}$. Then $\mathbb{C}^{\mathcal{C}}$ is an inner product space.

Definition 4.6 (Fourier Analysis)**.** For a coloring $U\in\mathcal{C}$, we define the $\mathcal{C}$-vector

$$\chi_U(C) := \omega^{\sum_{v\in V} U(v)C(v)},$$

where $\omega := \omega_{k-1} := \mathrm{e}^{2\pi i/(k-1)}$ is the primitive $(k-1)^{\text{st}}$ root of unity. The collection $\{\chi_U\}_{U\in\mathcal{C}}$ forms an orthonormal basis for $\mathbb{C}^{\mathcal{C}}$, called the Fourier Basis. We write $\hat{g}(U) := \langle g,\chi_U\rangle$ to denote the Fourier coefficient of g at "frequency" U. We say that the *support* of U is *supp*$(U) := \{v\in V : U(v)\neq 0\}$ for $U\in\mathcal{C}$.

Definition 4.7 (Reachability)**.** Fix an mCLIQUE network G'. For $a'\in V'$, let $R_{a'}(C) :=$ TRUE if the extremal NO-instance $G(C)$ can reach a'.[22] The Boolean vector $R_{a'}$ is identified with the $\mathcal{C}$-vector $R_{a'}(C) := [\![C \text{ can reach } a']\!]$.

Proposition 4.8 (Adjacent Reachability)**.** *If a' and b' are connected by an edge labeled $\{u,v\}$, where $\{u,v\}\in G(C)$, then $R_{a'}(C) = R_{b'}(C)$.*

Let $\mathbf{1}$ denote the all-ones $\mathcal{C}$-vector: $\mathbf{1}(C) = 1$ for all $C\in\mathcal{C}$; and similarly $\mathbf{0}$ the all-zeros $\mathcal{C}$-vector. Note that $R_{s'} = \mathbf{1}$; and if G' is sound (i. e., does not accept NO-instances), then $R_{t'} = \mathbf{0}$.

4.2 Invariant cover for cliques

This subsection adapts the framework of invariant cover from GEN to CLIQUE as Lemma 4.13, by suitably changing the definitions of an invariant cover (Definitions 4.9 and 4.10) and Fourier support (Definitions 4.11 and 4.12).

Definition 4.9 (l-Invariant)**.** For a literal l and an mCLIQUE network G', a $\mathcal{C}$-vector g is *l-invariant on G'* if for any a' and b' in $V(G')$ connected by an edge labeled l, $\langle R_{a'},g\rangle = \langle R_{b'},g\rangle$. We say that g is *l-invariant* if g is l-invariant on G' for any mCLIQUE network G'.

Definition 4.10 (Invariant Cover)**.** Consider a YES instance P. A collection of $\mathcal{C}$-vectors $\{g_{P,\iota}\}_{\iota\in I}$ over some index set I forms an *invariant cover* (for P) if (1) for any positive literal $l\in G(P)$, there is an $\iota\in I$ so that $g_{P,\iota}$ is l-invariant; and (2) for any $\iota\in I$, we have $\langle\mathbf{1},g_{P,\iota}\rangle = 1$.

Definition 4.11 (Fourier Support)**.** For a k-clique P, say a $\mathcal{C}$-vector g is *P-supported* if

1. g depends only on coloring in P, i. e., $g(C_1) = g(C_2)$ if $C_1(v) = C_2(v)$ for all $v\in V(P)$; or *equivalently*

[22] Hence $R_{a'}$ is the truth table at node a' restricted to the extremal NO-instances $G(\mathcal{C})$.

2. $\hat{g}(U) \neq 0$ only when $supp(U) \subseteq V(P)$.

Definition 4.12 (High Frequency Support)**.** We introduce *cut-off degree* and *agreement degree* for describing the Fourier support.

1. A $\mathcal{C}$-vector g has *cut-off degree* k if $\hat{g}(U) \neq 0$ only when $|supp(U)| \geq k$.
2. A collection of $\mathcal{C}$-vectors $\{g_\iota\}_{\iota \in I}$ *agrees up to degree* k if for all $U \in \mathcal{C}$ where $|supp(U)| \leq k$, we have $\hat{g}_{\iota_1}(U) = \hat{g}_{\iota_2}(U)$ for $\iota_1, \iota_2 \in I$.

Lemma 4.13 (Size Lower Bound from Invariants)**.** *For $3 \leq k \leq n^{1/100}$, if there is an invariant cover $\{g_{P,\iota}\}_{\iota \in I}$ such that (1) $g_{P,\iota}$ is P-supported, (2) $\{g_{P,\iota}\}_{\iota \in I}$ agrees up to degree $k-2$, and (3) $\|g_{P,\iota}\| \leq k^k$; then any sound m*CLIQUE *network G' having n' nodes for* CLIQUE *of size n and accepting all k-cliques, satisfies*

$$n' \geq (1/2)n^{k/100} = n^{\Theta(k)}.$$

Proof. Fix a k-clique P. Its YES-instance $G(P)$ is accepted by G' via some path P' labeled with edges from $G(P)$. Lemma 3.24 gives a node b'_P and a $\mathcal{C}$-vector $g_P := g_{P,\iota_1} - g_{P,\iota_2}$ such that $|\langle R_{b'_P}, g_P\rangle| \geq 1/n'$.[23] Note that $\|g_P\| \leq \|g_{P,u}\| + \|g_{P,v}\| \leq 2k^k$. Also, g_P has cut-off degree $k-1$ by Proposition 3.28, and is P-supported.

Let $q := n^{k/10}$, then Lemma 2.2 gives q sets $Q_1, Q_2, \ldots, Q_q \subseteq [N]$ satisfying $|Q_i| = k$ and $|Q_i \cap Q_j| \leq k-2$ for $1 \leq i \neq j \leq q$, where

$$N = \exp\left(\frac{k \ln n}{10(k-2)} + 1\right)\frac{k^2}{k-2} \leq n^{1/3}k^2 \ll n.$$

For each Q_i, construct the k-clique P_i with $V(P_i) = Q_i$, then the previous paragraph gives $b'_i := b'_{P_i}$ and $g_i := g_{P_i}$. Normalize $\tilde{g}_i := g_i/\|g_i\|$, then $\{\tilde{g}_i\}_{1 \leq i \leq q}$ is orthonormal (having disjoint Fourier support). Note that $1 \geq \|R_{b'}\|^2 \geq \sum_i |\langle R_{b'}, \tilde{g}_i\rangle|^2$ for any $b' \in V'$. Now

$$n' = \sum_{a' \in V'} 1 \geq \sum_i \sum_{a' \in V'} |\langle R_{a'}, \tilde{g}_i\rangle|^2 \geq \sum_i |\langle R_{b'_i}, \tilde{g}_i\rangle|^2 \geq q\left(\frac{1}{n'2k^k}\right)^2,$$

since

$$|\langle R_{b'_i}, \tilde{g}_i\rangle| \geq \frac{1}{n'}\frac{1}{\|g_i\|} \geq \frac{1}{n'2k^k}.$$

It follows that

$$(n')^3 \geq \frac{q}{4k^{2k}} = \frac{n^{k/10}}{4k^{2k}} \geq \frac{1}{4}n^{k/20}.$$ □

[23] Strictly speaking, $\mathcal{C}$-vectors in Section 4.1 are complex vectors while $\mathcal{C}$-vectors in Section 3.3.1 are real vectors. We may redefine $\mathcal{C}$-vectors in Section 3.3.1 as complex vectors. In doing so, the only necessary change is the proof of Lemma 3.24: we may apply Lemma 3.25 instead with $d_{j,\iota} :=$ real part of $\langle R_{v'_j}, g_{P,\iota}\rangle$.

4.3 Constructing invariant cover

This subsection constructs an invariant cover for CLIQUE (Lemma 4.14) by exploiting the symmetry in the problem. The lower bound for CLIQUE is proved as Theorem 4.19.

Fix a collection of vertices $\mathfrak{D}$ (e. g., $V(P) \cong [k]$) and a color set $\mathfrak{R}$ (e. g., $[k-1]$). A coloring $C\colon \mathfrak{D} \to \mathfrak{R}$ has a *monopoly* if exactly one color class has more than one vertex, i. e., $\left|\left\{c \in \mathfrak{R}\colon |C^{-1}(c)| > 1\right\}\right| = 1$, in which case the monopoly refers to vertices in $\mathfrak{D}$ with the dominating color. Later, for counting purposes, we will restrict attention to a subset of vertices in a clique, e. g., $\mathfrak{D} = [k-2]$ and $\mathfrak{R} = [k-1]$. In such cases, say C is *proper* if $C(u) = C(v)$ implies $u = v$ for $u, v \in \mathfrak{D}$.

Lemma 4.14 (Invariant Cover)**.** *Define the scaling factor $\phi(k) := (k-1)^{k-1}/(k-2)!(k-2)$. For any vertex v in a k-clique P, define a $\mathcal{C}$-vector $g_{P,v}$ by*

$$g_{P,v}(C) := \begin{cases} \phi(k)(-1)^{\tau}(\tau-2)! & \textit{if } C \textit{ (restricted to } P\textit{) has a monopoly with } \tau \textit{ vertices} \\ & \textit{and the monopoly does } \text{not } \textit{contain } v, \\ 0 & \textit{otherwise.} \end{cases}$$

Then (1) $g_{P,v}$ is P-supported, (2) $g_{P,v}$ is l-invariant when $l = \{u, v\}$ for any $u \in V(P) \setminus \{v\}$, (3) $\langle \mathbf{1}, g_{P,v}\rangle = 1$, (4) $\{g_{P,v}\}_{v \in P}$ agrees up to degree $k-2$, and (5) $\|g_{P,v}\| \le k^{k/2}$.

Proof. Item (1) is clear. For (2), note that if $g_{P,v}(C) \neq 0$ then C gives a unique color to v in P, i. e., $C(u) \neq C(v)$ for all $u \in V(P) \setminus \{v\}$. Hence if a' and b' are connected by an edge labeled $l = \{u, v\}$ where $u \in V(P) \setminus \{v\}$, then $R_{a'}(C) g_{P,v}(C) = R_{b'}(C) g_{P,v}(C)$ for any C, because either (i) $g_{P,v}(C) = 0$, or (ii) $C(u) \neq C(v)$, then $\{u, v\} \in G(C)$ and $R_{a'}(C) = R_{b'}(C)$ (Proposition 4.8).

For (3), note that when C gives v a unique color (denoted by c_v below) in P, ignoring v and its color reduces to a coloring $\tilde{C}\colon V(P) \setminus \{v\} \to [k-1] \setminus \left\{C(v)\right\}$, which is identified with $\tilde{C}\colon [k-1] \to [k-2]$. For $\tau \ge 2$, the number of such $\tilde{C}$ having a monopoly of size τ is

$$\binom{k-1}{\tau} \frac{(k-2)!}{(\tau-2)!},$$

hence

$$\begin{aligned} \langle \mathbf{1}, g_{P,v}\rangle = |\mathcal{C}|^{-1} \sum_{C \in \mathcal{C}} g_{P,v}(C) &= \frac{1}{(k-1)^k} \sum_{c_v \in [k-1]} \sum_{2 \le \tau \le k-1} \binom{k-1}{\tau} \frac{(k-2)!}{(\tau-2)!} \cdot \phi(k)(-1)^{\tau}(\tau-2)! \\ &= \frac{1}{k-2} \sum_{2 \le \tau \le k-1} \binom{k-1}{\tau} (-1)^{\tau} = 1, \end{aligned}$$

using the combinatorial identity $\sum_{0 \le \tau \le t} \binom{t}{\tau}(-1)^{\tau} = 0$ for $t \ge 1$.[24]

For (4), note that $\mathcal{C}$-vectors g_1 and g_2 agree up to degree $k-2$ if they "depend the same way on colorings whose domain has size at most $k-2$" (Claim 4.15). More precisely, introduce the following.

[24] Elchanan Mossel noticed that this implicit use of the principle of inclusion-exclusion for controlling Fourier coefficients is similar to the Efron-Stein decomposition [20].

For nesting vertex sets $\mathfrak{D}_1 \supseteq \mathfrak{D}_2$, a coloring $C_1 \colon \mathfrak{D}_1 \to \mathfrak{R}$ *extends* a coloring $C_2 \colon \mathfrak{D}_2 \to \mathfrak{R}$ if C_1 agrees with C_2 when restricted to $\mathfrak{D}_2$, i. e., $C_1(v) = C_2(v)$ for $v \in \mathfrak{D}_2$. Define

$$Ext(C_2, \mathfrak{D}_1) := \{C_1 \colon \mathfrak{D}_1 \to \mathfrak{R} \text{ extending } C_2\}$$

as all extensions of C_2 to $\mathfrak{D}_1$. The C_2-slice of a $\mathcal{C}$-vector g is $slice(g, C_2) := |E|^{-1} \sum_{C_1 \in E} g(C_1)$, where $E := Ext(C_2, \mathfrak{D}_1)$ with an appropriate $\mathfrak{D}_1$ (which will be clear from the context, e. g., V or $V(P)$).

Claim 4.15 (Agreement)**.** *$\mathcal{C}$-vectors g_1 and g_2 agree up to degree $k-2$ if $slice(g_1, C_2) = slice(g_2, C_2)$ for any coloring $C_2 \colon \mathfrak{D}_2 \to [k-1]$ such that $|\mathfrak{D}_2| \le k-2$.*

Proof. Given any coloring $U \in \mathcal{C}$ as frequency, partition $\mathcal{C}$ by colors over $supp(U)$, and rewrite

$$\hat{g}(U) = |\mathcal{C}|^{-1} \sum_{C \in \mathcal{C}} g(C) \overline{\chi_U}(C) = |\mathcal{S}|^{-1} \sum_{S \in \mathcal{S}} slice\big(g, S\big) \overline{\chi_U}(S),$$

where $\mathcal{S} := \big\{S \colon supp(U) \to [k-1]\big\}$ is the collection of colorings over the support of U (note that χ_U depends only on colors over $supp(U)$). Hence if $|supp(U)| \le k-2$, then

$$\hat{g}_1(U) = |\mathcal{S}|^{-1} \sum_{S \in \mathcal{S}} slice\big(g_1, S\big) \overline{\chi_U}(S) = |\mathcal{S}|^{-1} \sum_{S \in \mathcal{S}} slice\big(g_2, S\big) \overline{\chi_U}(S) = \hat{g}_2(U). \qquad \square$$

Hence (4) follows, if for any $u, v \in V(P)$, we have $slice(g_{P,u}, C_2) = slice(g_{P,v}, C_2)$ for any coloring $C_2 \colon \mathfrak{D}_2 \to [k-1]$ with $|\mathfrak{D}_2| \le k-2$. Further, we can assume $\mathfrak{D}_2 \subseteq V(P)$, since $g_{P,u}$ and $g_{P,v}$ are P-supported. Therefore, after computing the slice function (Claim 4.16) to show the independence of v (Claim 4.17), item (4) follows from Claim 4.18 (due to Claim 4.15).

Claim 4.16 (Indicator of Proper Coloring)**.** *If $\mathfrak{D}_2 \subseteq V(P)$ has size $|\mathfrak{D}_2| = k-1$, and $\mathfrak{D}_2 \ni v$, then for any coloring $C_2 \colon \mathfrak{D}_2 \to [k-1]$, we have*

$$slice(g_{P,v}, C_2) = \frac{k-2}{k-1} \phi(k) [\![C_2 \textit{ is proper}]\!] .$$

Proof. Let $E := Ext\big(C_2, V(P)\big)$, then any extension $C_1 \in E$ colors one more vertex $w \in V(P) \setminus \mathfrak{D}_2$ than C_2, and

$$slice(g_{P,v}, C_2) = \frac{1}{k-1} \sum_{C_1 \in E} slice(g_{P,v}, C_1).$$

If C_2 is proper, then by considering the coloring of w,

1. one extension $C_1 \in E$ has a monopoly of size 2 containing v, where $slice(g_{P,v}, C_1) = 0$; and

2. $k-2$ extensions $C_1 \in E$ has a monopoly of size 2 *not* containing v, where $slice(g_{P,v}, C_1) = \phi(k)$ as $(-1)^2(2-2)! = 1$.

Thus

$$slice(g_{P,v}, C_2) = \frac{k-2}{k-1} \phi(k)$$

if C_2 is proper.

Otherwise, C_2 is not proper, and has a color class with more than one vertex. If C_2 does *not* have a monopoly or the monopoly contains v, then any extension $C \in Ext(C_2, V)$ has $g_{P,v}(C) = 0$, hence $slice(g_{P,v}, C_2) = 0$.

For the remaining case, C_2 has a monopoly of size $\tau \geq 2$ *not* containing v. By considering the coloring of w,

- one extension $C_1 \in E$ has a monopoly of size $\tau + 1$ *not* containing v, where $slice(g_{P,v}, C_1) = \phi(k)(-1)^{\tau+1}(\tau-1)!$; and
- $\tau - 1$ extensions $C_1 \in E$ have a monopoly of size τ *not* containing v, where $slice(g_{P,v}, C_1) = \phi(k)(-1)^{\tau}(\tau-2)!$.

Hence

$$slice(g_{P,v}, C_2) = \frac{\phi(k)}{k-1}\big((-1)^{\tau+1}(\tau-1)! + (\tau-1)(-1)^{\tau}(\tau-2)!\big) = 0\,. \qquad \square$$

Claim 4.17 (Independence of v). *If $\mathfrak{D}_2 \subseteq V(P)$ has size $|\mathfrak{D}_2| = k-2$, then for any coloring $C_2 \colon \mathfrak{D}_2 \to [k-1]$, we have*

$$slice(g_{P,v}, C_2) = \frac{k-2}{(k-1)^2}\phi(k)[\![C_2 \text{ is proper}]\!]\,,$$

which is independent of v.

Proof. Let $\mathfrak{D}_1$ be an arbitrary subset such that $\mathfrak{D}_1 \subseteq V(P)$ and $|\mathfrak{D}_1| = k-1$ and $\mathfrak{D}_1 \supseteq \mathfrak{D}_2 \cup \{v\}$. Now $slice(g_{P,v}, C_2) = |E|^{-1}\sum_{C_1 \in E} slice(g_{P,v}, C_1)$ where $E := Ext(C_2, \mathfrak{D}_1)$. If C_2 is *not* proper, neither is any $C_1 \in E$, hence $slice(g_{P,v}, C_1) = 0$ by Claim 4.16, and $slice(g_{P,v}, C_2) = 0$. Otherwise C_2 is proper, then exactly one extension $C_1 \in E$ is proper, and the result follows from Claim 4.16. $\square$

Claim 4.18 (Equality of Slices). *For any $C_2 \colon \mathfrak{D}_2 \to [k-1]$ with $\mathfrak{D}_2 \subseteq V(P)$ and $|\mathfrak{D}_2| \leq k-2$, we have $slice(g_{P,u}, C_2) = slice(g_{P,v}, C_2)$ for any $u, v \in V(P)$.*

Proof. Let $\mathfrak{D}_1$ be an arbitrary subset of size $|\mathfrak{D}_1| = k-2$ such that $\mathfrak{D}_2 \subseteq \mathfrak{D}_1 \subseteq V(P)$. Let $E := Ext(C_2, \mathfrak{D}_1)$, then Claim 4.17 gives $slice(g_{P,u}, C_1) = slice(g_{P,v}, C_1)$ for $C_1 \in E$, so

$$slice(g_{P,u}, C_2) = |E|^{-1}\sum_{C_1 \in E} slice(g_{P,u}, C_1) = |E|^{-1}\sum_{C_1 \in E} slice(g_{P,v}, C_1) = slice(g_{P,v}, C_2)\,. \qquad \square$$

To compute $\|g_{P,v}\|$ for (5), partition $\mathcal{C}$ by coloring to v (denoted by c_v below),

$$\begin{aligned}\langle g_{P,v}, g_{P,v}\rangle = |\mathcal{C}|^{-1}\sum_{C \in \mathcal{C}} g_{P,v}(C)^2 &= \frac{1}{(k-1)^k}\sum_{c_v \in [k-1]}\sum_{2 \leq \tau \leq k-1}\binom{k-1}{\tau}\frac{(k-2)!}{(\tau-2)!}\cdot\big(\phi(k)(-1)^{\tau}(\tau-2)!\big)^2 \\ &= \frac{(k-1)^{k-1}}{(k-2)^2}\sum_{2 \leq \tau \leq k-1}\binom{k-1}{\tau}\frac{(\tau-2)!}{(k-2)!} \leq (k-1)^k\,,\end{aligned}$$

since

$$\binom{k-1}{\tau}\frac{(\tau-2)!}{(k-2)!} \leq k-1\,.$$

So $\|g_{P,v}\| \leq k^{k/2}$. $\square$

Theorem 4.19 (Size Lower Bound)**.** *For $3 \leq k \leq n^{1/100}$, any m*CLIQUE *network G' having n' nodes and computing k-*CLIQUE *of size n, satisfies*

$$n' \geq \frac{1}{2} n^{k/100} = n^{\Theta(k)} .$$

Proof. Use Lemmas 4.13 and 4.14. □

Acknowledgements

We thank Yuval Filmus and Robert Robere for pointing out an error in an earlier version, and for substantially simplifying the conceptual framework with a backward induction of Lemma 3.35. The first-named author thanks Anand Bhaskar, Stephen Cook, Pierre McKenzie, Thomas Watson, and Dustin Wehr for helpful discussions. We also thank the anonymous reviewers of STOC'12 and the anonymous reviewers of ToC and their colleagues, for valuable comments on presentations.

References

[1] MIKLÓS AJTAI: A non-linear time lower bound for Boolean branching programs. *Theory of Computing*, 1(8):149–176, 2005. Preliminary version in FOCS'99. [doi:10.4086/toc.2005.v001a008] 392

[2] MIKLÓS AJTAI, LÁSZLÓ BABAI, PÉTER HAJNAL, JÁNOS KOMLÓS, PAVEL PUDLÁK, VOJTECH RÖDL, ENDRE SZEMERÉDI, AND GYÖRGY TURÁN: Two lower bounds for branching programs. In *Proc. 18th STOC*, pp. 30–38. ACM Press, 1986. [doi:10.1145/12130.12134] 392

[3] NOGA ALON AND RAVI B. BOPPANA: The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987. [doi:10.1007/BF02579196] 390

[4] KAZUYUKI AMANO AND AKIRA MARUOKA: The potential of the approximation method. *SIAM J. Comput.*, 33(2):433–447, 2004. Preliminary version in FOCS'96. [doi:10.1137/S009753970138445X] 392

[5] ALEXANDER E. ANDREEV: On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Mathematics Doklady*, 31(3):530–534, 1985. 392

[6] DAVID A. MIX BARRINGTON AND PIERRE MCKENZIE: Oracle branching programs and Logspace versus P. *Inform. and Comput.*, 95(1):96–115, 1991. Preliminary version in MFCS'89. [doi:10.1016/0890-5401(91)90017-V] 394

[7] DAVID A. MIX BARRINGTON AND HOWARD STRAUBING: Superlinear lower bounds for bounded-width branching programs. *J. Comput. System Sci.*, 50(3):374–381, 1995. Preliminary version in SCT'91. [doi:10.1006/jcss.1995.1029] 392

[8] PAUL BEAME, THATHACHAR S. JAYRAM, AND MICHAEL E. SAKS: Time-space tradeoffs for branching programs. *J. Comput. System Sci.*, 63(4):542–572, 2001. Preliminary version in FOCS'98. [doi:10.1006/jcss.2001.1778] 392

[9] PAUL BEAME, MICHAEL E. SAKS, XIAODONG SUN, AND ERIK VEE: Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50(2):154–195, 2003. Preliminary version in FOCS'00. [doi:10.1145/636865.636867] 392

[10] CHARLES H. BENNETT: Time/space trade-offs for reversible computation. *SIAM J. Comput.*, 18(4):766–776, 1989. [doi:10.1137/0218053] 394

[11] CHRISTER BERG AND STAFFAN ULFBERG: Symmetric approximation arguments for monotone lower bounds without sunflowers. *Comput. Complexity*, 8(1):1–20, 1999. [doi:10.1007/s000370050017] 392

[12] MARÍA LUISA BONET, JUAN LUIS ESTEBAN, NICOLA GALESI, AND JAN JOHANNSEN: On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.*, 30(5):1462–1484, 2000. Preliminary versions in FOCS'98 and ECCC. [doi:10.1137/S0097539799352474] 392

[13] MARÍA LUISA BONET, TONIANN PITASSI, AND RAN RAZ: Lower bounds for cutting planes proofs with small coefficients. *J. Symbolic Logic*, 62(3):708–728, 1997. Preliminary version in STOC'95. JSTOR. 392

[14] RAVI B. BOPPANA: Threshold functions and bounded depth monotone circuits. *J. Comput. System Sci.*, 32(2):222–229, 1986. Preliminary version in STOC'84. [doi:10.1016/0022-0000(86)90027-9] 392

[15] RAVI B. BOPPANA AND MICHAEL SIPSER: The complexity of finite functions. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pp. 757–804. Elsevier and MIT Press, 1990. [ACM:114886] 391, 392

[16] ALLAN BORODIN: On relating time and space to size and depth. *SIAM J. Comput.*, 6(4):733–744, 1977. [doi:10.1137/0206054] 390, 391

[17] STEPHEN A. COOK: An observation on time-storage trade off. *J. Comput. System Sci.*, 9(3):308–316, 1974. Preliminary version in STOC'73. [doi:10.1016/S0022-0000(74)80046-2] 394, 396, 397

[18] STEPHEN A. COOK, PIERRE MCKENZIE, DUSTIN WEHR, MARK BRAVERMAN, AND RAHUL SANTHANAM: Pebbles and branching programs for tree evaluation. *ACM Trans. Computation Theory*, 3(2):4:1–4:43, 2012. Preliminary versions in FSTTCS'09 and MFCS'09. [doi:10.1145/2077336.2077337] 392

[19] JEFF EDMONDS, RUSSELL IMPAGLIAZZO, STEVEN RUDICH, AND JIŘÍ SGALL: Communication complexity towards lower bounds on circuit depth. *Comput. Complexity*, 10(3):210–246, 2001. Preliminary version in FOCS'91. [doi:10.1007/s00037-001-8195-x] 391

[20] BRADLEY EFRON AND CHARLES STEIN: The jackknife estimate of variance. *Ann. Stat.*, 9(3):586–596, 1981. [doi:10.1214/aos/1176345462] 411

[21] ANNA GÁL, MICHAL KOUCKÝ, AND PIERRE MCKENZIE: Incremental branching programs. *Theory Comput. Syst.*, 43(2):159–184, 2008. Preliminary versions at Complexity of Boolean Functions and CSR'06. [doi:10.1007/s00224-007-9049-y] 392

[22] MIKAEL GOLDMANN AND JOHAN HÅSTAD: A simple lower bound for monotone clique using a communication game. *Inform. Process. Lett.*, 41(4):221–226, 1992. [doi:10.1016/0020-0190(92)90184-W] 391

[23] MIKAEL GOLDMANN AND JOHAN HÅSTAD: Monotone circuits for connectivity have depth $(\log n)^{2-o(1)}$. *SIAM J. Comput.*, 27(5):1283–1294, 1998. Preliminary version in STOC'95. [doi:10.1137/S0097539795285631] 392

[24] MICHELANGELO GRIGNI: *Structure in Monotone Complexity*. Ph. D. thesis, Massachusetts Institute of Technology, 1991. Available at CiteSeerX. [ACM:918937] 391, 392

[25] MICHELANGELO GRIGNI AND MICHAEL SIPSER: Monotone separation of logarithmic space from logarithmic depth. *J. Comput. System Sci.*, 50(3):433–437, 1995. Preliminary version in SCT'91. [doi:10.1006/jcss.1995.1033] 391

[26] ARMIN HAKEN: Counting bottlenecks to show monotone $\mathsf{P} \neq \mathsf{NP}$. In *Proc. 36th FOCS*, pp. 36–40. IEEE Comp. Soc. Press, 1995. [doi:10.1109/SFCS.1995.492460] 390

[27] DANNY HARNIK AND RAN RAZ: Higher lower bounds on monotone size. In *Proc. 32nd STOC*, pp. 378–387. ACM Press, 2000. Full version available at author's home page. [doi:10.1145/335305.335349] 392

[28] RUSSELL IMPAGLIAZZO, TONIANN PITASSI, AND ALASDAIR URQUHART: Upper and lower bounds for tree-like cutting planes proofs. In *Proc. 9th Ann. Symp. on Logic in Computer Science (LICS'94)*, pp. 220–228. IEEE Comp. Soc. Press, 1994. [doi:10.1109/LICS.1994.316069] 392

[29] JAN JOHANNSEN: Depth lower bounds for monotone semi-unbounded fan-in circuits. *RAIRO - Theoretical Informatics and Applications*, 35(3):277–286, 2001. [doi:10.1051/ita:2001120] 391

[30] NEIL D. JONES AND WILLIAM T. LAASER: Complete problems for deterministic polynomial time. *Theor. Comput. Sci.*, 3(1):105–117, 1976. Preliminary version in STOC'74. [doi:10.1016/0304-3975(76)90068-2] 394

[31] STASYS JUKNA: Finite limits and monotone computations: The lower bounds criterion. In *Proc. 12th IEEE Conf. on Computational Complexity (CCC'97)*, pp. 302–313, 1997. [doi:10.1109/CCC.1997.612325] 392

[32] MAURICIO KARCHMER: On proving lower bounds for circuit size. In *Proc. 8th IEEE Conf. on Structure in Complexity Theory (SCT'93)*, pp. 112–118. IEEE Comp. Soc. Press, 1993. [doi:10.1109/SCT.1993.336535] 392

[33] MAURICIO KARCHMER, RAN RAZ, AND AVI WIGDERSON: Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Comput. Complexity*, 5(3-4):191–204, 1995. Preliminary version in SCT'91. [doi:10.1007/BF01206317] 391

[34] MAURICIO KARCHMER AND AVI WIGDERSON: Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discrete Math.*, 3(2):255–265, 1990. Preliminary version in STOC'88. [doi:10.1137/0403021] 390, 391

[35] MAURICIO KARCHMER AND AVI WIGDERSON: On span programs. In *Proc. 8th IEEE Conf. on Structure in Complexity Theory (SCT'93)*, pp. 102–111, 1993. [doi:10.1109/SCT.1993.336536] 392

[36] MARIA M. KLAWE: A tight bound for black and white pebbles on the pyramid. *J. ACM*, 32(1):218–228, 1985. Preliminary version in FOCS'83. [doi:10.1145/2455.214115] 396

[37] JAN KRAJÍČEK: Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symbolic Logic*, 62(2):457–486, 1997. JSTOR. 392

[38] RICHARD KRÁLOVIC: Time and space complexity of reversible pebbling. *RAIRO - Theoretical Informatics and Applications*, 38(2):137–161, 2004. Preliminary version in SOFSEM'01. [doi:10.1051/ita:2004008] 394

[39] MATTHIAS KRAUSE: Lower bounds for depth-restricted branching programs. *Inform. and Comput.*, 91(1):1–14, 1991. [doi:10.1016/0890-5401(91)90072-A] 392

[40] KLAUS-JÖRN LANGE, PIERRE MCKENZIE, AND ALAIN TAPP: Reversible space equals deterministic space. *J. Comput. System Sci.*, 60(2):354–367, 2000. Preliminary version in CCC'97. [doi:10.1006/jcss.1999.1672] 393

[41] KETAN MULMULEY: Lower bounds in a parallel model without bit operations. *SIAM J. Comput.*, 28(4):1460–1509, 1999. [doi:10.1137/S0097539794282930] 392

[42] KATSUTOSHI NAKAYAMA AND AKIRA MARUOKA: Loop circuits and their relation to Razborov's approximation model. *Inform. and Comput.*, 119(2):154–159, 1995. [doi:10.1006/inco.1995.1083] 392

[43] ÈDUARD IVANOVIČ NEČIPORUK: On a Boolean function. *Soviet Mathematics Doklady*, 7(4):999–1000, 1966. 392

[44] NOAM NISAN AND AVI WIGDERSON: Hardness vs randomness. *J. Comput. System Sci.*, 49(2):149–167, 1994. [doi:10.1016/S0022-0000(05)80043-1] 393

[45] AARON POTECHIN: Bounds on monotone switching networks for directed connectivity. In *Proc. 51st FOCS*, pp. 553–562. IEEE Comp. Soc. Press, 2010. An updated version to appear in Journal of the ACM. [doi:10.1109/FOCS.2010.58] 391, 392, 393, 394, 396, 397, 398

[46] RAN RAZ AND PIERRE MCKENZIE: Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. Preliminary version in FOCS'97. [doi:10.1007/s004930050062] 390, 391, 392, 396, 398, 408

[47] RAN RAZ AND AVI WIGDERSON: Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. Preliminary version in STOC'90. [doi:10.1145/146637.146684] 391

[48] ALEXANDER A. RAZBOROV: Lower bounds for the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady*, 31(2):354–357, 1985. 390, 408

[49] ALEXANDER A. RAZBOROV: Lower bounds on monotone complexity of the logical permanent. *Mathematical Notes of the Academy of Sciences of the USSR*, 37(6):485–493, 1985. [doi:10.1007/BF01157687] 392

[50] ALEXANDER A. RAZBOROV: On the method of approximations. In *Proc. 21st STOC*, pp. 167–176. ACM Press, 1989. [doi:10.1145/73007.73023] 392

[51] ALEXANDER A. RAZBOROV: Lower bounds of the complexity of symmetric Boolean functions of contact-rectifier circuits. *Mathematical Notes of the Academy of Sciences of the USSR*, 48(6):1226–1234, 1990. [doi:10.1007/BF01240265] 392

[52] ALEXANDER A. RAZBOROV: Lower bounds for deterministic and nondeterministic branching programs. In *8th Internat. Symp. on Fundamentals of Computation Theory, 8th International Symposium (FCT'91)*, pp. 47–60, 1991. [doi:10.1007/3-540-54458-5_49] 391, 392, 393

[53] ALEXANDER A. RAZBOROV, AVI WIGDERSON, AND ANDREW CHI-CHIH YAO: Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. *Combinatorica*, 22(4):555–574, 2002. Preliminary version in STOC'97. [doi:10.1007/s00493-002-0007-7] 392

[54] OMER REINGOLD: Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, 2008. Preliminary version in STOC'05. [doi:10.1145/1391289.1391291] 391, 392, 393

[55] BENJAMIN ROSSMAN: The monotone complexity of *k*-clique on random graphs. In *Proc. 51st FOCS*, pp. 193–201. IEEE Comp. Soc. Press, 2010. [doi:10.1109/FOCS.2010.26] 392, 408

[56] JANOS SIMON AND SHI-CHUN TSAI: On the bottleneck counting argument. *Theoret. Comput. Sci.*, 237(1-2):429–437, 2000. Preliminary version in CCC'97. [doi:10.1016/S0304-3975(99)00321-7] 392

[57] ÉVA TARDOS: The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. [doi:10.1007/BF02122563] 392

[58] LUCA TREVISAN: Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001. Preliminary version in STOC'99. [doi:10.1145/502090.502099] 393

[59] DUSTIN WEHR: Pebbling and branching programs solving the tree evaluation problem. Technical report, 2010. [arXiv:1002.4676] 392

[60] DUSTIN WEHR: Lower bound for deterministic semantic-incremental branching programs solving GEN. Technical report, 2011. [arXiv:1101.2705] 392

[61] ANDREW CHI-CHIH YAO: Circuits and local computation. In *Proc. 21st STOC*, pp. 186–196. ACM Press, 1989. [doi:10.1145/73007.73025] 392

[62] ANDREW CHI-CHIH YAO: A lower bound for the monotone depth of connectivity. In *Proc. 35th FOCS*, pp. 302–308. IEEE Comp. Soc. Press, 1994. [doi:10.1109/SFCS.1994.365685] 392

AUTHORS

Siu Man Chan
Postdoc Department of Computer Science
University of Toronto
siuman@cs.utoronto.ca

Aaron Potechin
Ph. D. student
Mathematics Department
Massachusetts Institute of Technology
aaron@potechin.org

ABOUT THE AUTHORS

SIU MAN CHAN is a postdoc currently at Toronto, after spending one year at Princeton. He completed his Ph. D. at Berkeley under Luca Trevisan and Elchanan Mossel. He did his M. S. at Toronto under Toniann Pitassi, and his undergrad at the Chinese University of Hong Kong under Leizhen Cai. He is interested in lower bounds in computational complexity, like his clone Siu On. Siu Man is currently occupied with space and parallel complexity, and more generally with the understanding of proofs, algorithms, and complexity in different mathematical models via combinatorics. He procrastinates by dreaming of upgrading his camera gears and buying more lenses.

AARON POTECHIN did his undergraduate studies at Princeton, which was followed by Part III of the Mathematical Tripos at Cambridge. He is now a graduate student at MIT working under Jonathan Kelner. While he is interested in a wide variety of topics, his primary research focus for the past few years has been on trying to prove space lower bounds using the switching network model. At FOCS 2010 he received the Machtey Award for the Best Student Paper for his paper "Bounds on Monotone Switching Networks for Directed Connectivity."