Fractional coverings, greedy coverings, and rectifier networks

Dmitry Chistikov[1], Szabolcs Iván[2], Anna Lubiw[3], and Jeffrey Shallit[3]

1 **Max Planck Institute for Software Systems (MPI-SWS), Germany,** dch@mpi-sws.org
2 **University of Szeged, Hungary,** szabivan@inf.u-szeged.hu
3 **School of Computer Science, University of Waterloo, Canada,** {alubiw,shallit}@cs.uwaterloo.ca

Abstract

A rectifier network is a directed acyclic graph with distinguished sources and sinks; it is said to compute a Boolean matrix M that has a 1 in the entry (i, j) iff there is a path from the jth source to the ith sink. The smallest number of edges in a rectifier network that computes M is a classic complexity measure on matrices, which has been studied for more than half a century.

We explore two well-known techniques that have hitherto found little to no applications in this theory. Both of them build upon a basic fact that depth-2 rectifier networks are essentially weighted coverings of Boolean matrices with rectangles. We obtain new results by using *fractional* and *greedy* coverings (defined in the standard way).

First, we show that all *fractional* coverings of the so-called full triangular matrix have cost at least $n \log n$. This provides (a fortiori) a new proof of the tight lower bound on its depth-2 complexity (the exact value has been known since 1965, but previous proofs are based on different arguments). Second, we show that the *greedy* heuristic is instrumental in tightening the upper bound on the depth-2 complexity of the Kneser-Sierpiński (disjointness) matrix. The previous upper bound is $O(n^{1.28})$, and we improve it to $O(n^{1.17})$, while the best known lower bound is $\Omega(n^{1.16})$. Third, using *fractional* coverings, we obtain a form of direct product theorem that gives a lower bound on unbounded-depth complexity of Kronecker (tensor) products of matrices. In this case, the *greedy* heuristic shows (by an argument due to Lovász) that our result is only a logarithmic factor away from the "full" direct product theorem. Our second and third results constitute progress on open problem 7.3 and resolve, up to a logarithmic factor, open problem 7.5 from a recent book by Jukna and Sergeev (in Foundations and Trends in Theoretical Computer Science (2013)).

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Introduced in the 1950s, *rectifier networks* are one of the oldest and most basic models in the theory of computing. They are directed acyclic graphs with distinguished input and output nodes; a rectifier network is said to *compute* (or *express*) the Boolean matrix M that has a 1 in the entry (i, j) iff there is a path from the jth input to the ith output. Equivalently, rectifier networks can be viewed as Boolean circuits that consist entirely of OR gates of arbitrary fan-in. This simple model of computation has attracted a lot of attention [16], because it captures the "topological" core of other models: complexity bounds for rectifier networks extend in one way or another to Boolean circuits (i.e., circuits with Boolean gates) and to switching circuits [31, 27].

Given a matrix M, what is the smallest number of edges in a rectifier network that computes M? Denote this number by $\mathsf{OR}(M)$—this is a complexity measure on Boolean

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

matrices. This measure is fairly well understood: we know, from Nechiporuk [30], that the maximum of $\mathsf{OR}(M)$ grows as $n^2/2\log n$ as $n \to \infty$ if M is $n \times n$; we also know that random $n \times n$-matrices have complexity very close to $n^2/2\log n$. The "shape" of these two facts is reminiscent of the standard circuit complexity of Boolean functions over AND, OR, and NOT gates—but for them, the maximum is $2^n/n$ instead of $n^2/2\log n$.

However, much more is known about the measure $\mathsf{OR}(\cdot)$: there are explicit sequences of matrices that have complexity $n^{2-o(1)}$, close to the maximum (in contrast, for circuits over AND, OR, and NOT gates, exhibiting a single sequence of functions that require a superlinear number of gates would be a tremendous breakthrough). In fact, nowadays a range of methods are available for obtaining upper and lower bounds on $\mathsf{OR}(M)$ for specific matrices M; we refer the interested reader to the recent book by Jukna and Sergeev [16].

Many natural questions, however, remain open. Jukna and Sergeev list 19 open problems about $\mathsf{OR}(\cdot)$ and related complexity measures. Several of them refer to very restricted submodels, such as rectifier networks of depth 2: that is, networks where all paths contain (at most) 2 edges. A depth-2 rectifier network expressing a matrix M is essentially a *covering* of M—a collection of (rectangular) all-1 submatrices of M whose disjunction is M. In our work, we look into the corresponding complexity measure $\mathsf{OR}_2(\cdot)$ as well as $\mathsf{OR}(\cdot)$. We build upon the connection between rectifier networks and (weighted) set coverings and explore two well-known ideas that have previously found few applications in the study of rectifier networks: they are associated with fractional and greedy coverings respectively.

Fractional coverings are a generalization of usual set coverings. In the usual set cover problem, each set S can be either included or not included in the solution (i.e., in the covering); in the fractional version each set can be partially included: a solution assigns to each set S a real number $x_S \in [0;1]$, and for every element s of the universe the sum $\sum_{s \in S} x_S$ should be equal to or exceed 1. In other words, fractional coverings arise from linear relaxation of the integer program that expresses the set cover problem. *Greedy coverings* are, in contrast, usual coverings; they are the outcome of applying the standard greedy heuristic to an instance of the set cover problem: at each step, the algorithm picks a set S that covers the largest number of yet uncovered elements s. In our work, we use fractional and greedy coverings to obtain estimates on the values of $\mathsf{OR}_2(M)$ and $\mathsf{OR}(M)$.

Our results

First, we demonstrate that $\mathsf{OR}_2(T_n) = n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$, where T_n is the so-called full triangular matrix: an upper-triangular matrix that has 1s everywhere above the main diagonal and 0s on the diagonal and below. In this problem, the upper bound is easy and the challenge is to prove the lower bound. This was previously done by Krichevskii [20], and our paper provides a different proof of independent interest. In fact, we prove a stronger statement: all *fractional* coverings of T_n have large associated cost (Theorem 4). To this end, we take the linear program that expresses the fractional set cover problem and find a good feasible solution to the dual program. The value of this solution then gives a lower bound on the cost of all feasible solutions to the primal—that is, on the cost of fractional coverings. Since integral coverings are just a special case of fractional coverings, the result follows.

Second, we improve the upper bound on the value of $\mathsf{OR}_2(D_n)$, where D_n is the disjointness matrix, also known as the Kneser-Sierpiński matrix. This constitutes progress on open problem 7.3 in Jukna and Sergeev's book [16], where the previously known bounds are obtained. The previous upper bound is $O(n^{1.28})$, and our Theorem 8 improves it to $O(n^{1.17})$, while the best known lower bound is $\Omega(n^{1.16})$. To achieve this improvement, we subdivide the instance of the weighted set cover problem (in which the optimal value is $\mathsf{OR}_2(D_n)$) into polylog(n) natural subproblems and reduce them, by imposing an additional restriction, to instances of unweighted set cover problems. We then solve these instances with the *greedy* heuristic; the upper bound in the analysis invokes the so-called greedy covering lemma by Sapozhenko [34], also known as the Lovász–Stein theorem [23, 38]. This gives us the desired upper bound on $\mathsf{OR}_2(D_n)$; in fact, the greedy strategy turns out to be optimal, and the optimal exponent in $\mathsf{OR}_2(D_n)$ comes from a numerical optimization problem. As an intermediate result we determine, up to a polylogarithmic factor, the value of $\mathsf{OR}_2(D_k^m)$ where D_k^m is the adjacency matrix of the Kneser graph on $2\binom{k}{m}$ vertices.

Finally, we obtain (Theorem 13) a form of direct product theorem for the $\mathsf{OR}(\cdot)$ measure: $\mathsf{OR}(K \otimes M) \geq \mathsf{rk}_\vee^*(K) \cdot \mathsf{OR}(M)$. Here $K \otimes M$ denotes the Kronecker product of matrices K and M, and $\mathsf{rk}_\vee^*(K)$ is a fractional analogue of the Boolean rank of K. This resolves, up to a logarithmic factor, open problem 7.5 in the list of Jukna and Sergeev [16], which asks for the lower bound of $\mathsf{rk}_\vee(K) \cdot \mathsf{OR}(M)$ where $\mathsf{rk}_\vee(K) \geq \mathsf{rk}_\vee^*(K)$ is the Boolean rank of K. (In fact, a related question for unambiguous rectifier networks, or SUM-circuits, is originally due to Find et al. [6]; our technique applies to this model as well, giving an analogous inequality for the measure $\mathsf{SUM}(\cdot)$, see Corollary 15.) Suppose K is an $m \times n$ matrix; then, by the argument due to Lovász [24], the *greedy* heuristic shows that $\mathsf{rk}_\vee^*(K) \geq \mathsf{rk}_\vee(K)/(1 + \log mn)$, so our lower bound is indeed at most a logarithmic factor away from the "full" direct product theorem. To prove our lower bound, we take the linear programming formulation of the *fractional* set cover problem for the matrix K and use components of the optimal solution to the dual program to guide our argument. It is interesting to see how reasoning about coverings, or, equivalently, about depth-2 rectifier networks, enables us to obtain meaningful lower bounds on the size of rectifier networks that have unbounded depth.

2 Discussion and related work

We use the matrix language in this paper, but all results can be restated in terms of biclique coverings of bipartite graphs.

The **OR_2-complexity of full triangular matrices**, T_n, is tightly related to results on biclique coverings of complete undirected (non-bipartite) graphs from the early days of the theory of computing. The $n \log n$ lower bound, in one form or another, was known to Hansel [10], Krichevskii [20], Katona and Szemerédi [19], and Tarján [39].[1] Apart from purely combinatorial considerations, the interest in this problem is motivated by its applications in formula and switching-circuit complexity of the Boolean threshold-2 function (which takes on the value 1 if and only if at least two of its inputs are set to 1). For more context, see treatments by Radhakrishnan [33] and Lozhkin [26]. Our lower bound is obtained in a slightly more restrictive setting, because of explicit asymmetry: for $\mathsf{OR}_2(T_n)$, one needs to cover entries (i, j) with $i < j$ in the matrix; in biclique coverings of undirected graphs, it suffices to cover either of (i, j) and (j, i). Nevertheless, to the best of our knowledge, ours is the only proof that goes via linear programming (LP) duality and provides a tight lower bound on the size of *fractional* coverings. This result is new; we are not aware of other lower bounds for rectifier networks that come from feasible solutions to the LP dual (in approximation algorithms, a related technique is known under the name of "dual fitting" [44, Section 9.4]).

As for the **greedy heuristics**, we are not the first to use them in the context of depth-2 rectifier networks. Andreev [1] obtained a tight worst-case upper bound for a class of matrices potentially containing "wildcard" entries $(*)$. This upper bound is in terms of the number of

[1] Not all of these arguments compute the *exact* value of $\mathsf{OR}_2(T_n)$.

occurrences of 0s and 1s, provided that these numbers satisfy certain conditions as the matrix size tends to infinity. Our Theorem 8, however, does not follow from Andreev's worst-case bound. The disjointness matrix, D_n, which we apply this technique to, is a well-studied object in communication complexity [21]; it is a discrete version of the Sierpiński triangle. Boyar and Find [2] and Selezneva [35] proved that $\mathsf{OR}(D_n) = \Theta(n \log n)$ and $\mathsf{SUM}(D_n) = \frac{1}{2} n \log n$.[2] In depth 2, the previous bounds are due to Jukna and Sergeev [16]; it is unknown if greedy heuristics are also of use for SUM-circuits, as our upper bound for D_n does not extend to this model (our coverings are not partitions).

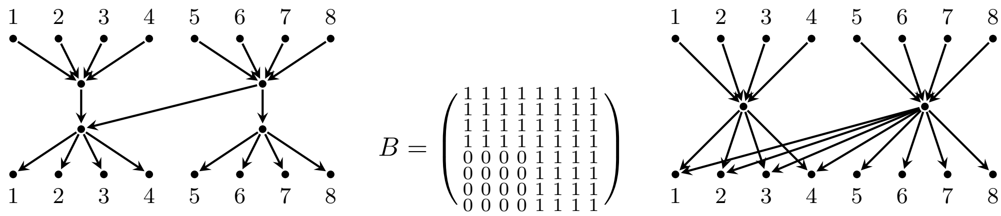
Direct sum and direct product theorems in the theory of computing are statements of the following form: when faced with several instances of the same problem on different independent inputs, there is no better strategy than solving each instance independently.[3] For rectifier networks, these questions are associated with the complexity of Kronecker (tensor) products of matrices. Indeed, denote the $k \times k$-identity matrix by I_k, then $I_k \otimes M$ is the block-diagonal matrix with k copies of M on the diagonal. It is not difficult to show that $\mathsf{OR}(I_k \otimes M) \geq k \cdot \mathsf{OR}(M)$, and a natural generalization asks whether $\mathsf{OR}(K \otimes M) \geq \mathsf{rk}_\vee(K) \cdot \mathsf{OR}(M)$ for any matrix K—see Find et al. [6] and Jukna and Sergeev [16, Sections 2.4, 3.6, and open problem 7.5]. To date, this inequality is only known to hold in special cases. For example, Find et al. [6] can show this lower bound when the matrix K has a fooling set of size $\mathsf{rk}_\vee(K)$; however, the size of the largest fooling set does not approximate the Boolean rank, as observed, e.g., by Gruber and Holzer [9] (they use the graph-theoretic language, with bipartite dimension instead of $\mathsf{rk}_\vee$). As another example, denote by $|M|$ the number of 1s in the matrix M and assume that M has no all-1 submatrices of size $(k+1) \times (l+1)$. Then the inequality $\mathsf{OR}(M) \geq |M|/kl$ is a well-known lower bound due to Nechiporuk [31], subsequently rediscovered by Mehlhorn [27], Pippenger [32], and Wegener [43]; Jukna and Sergeev [16, Theorem 3.20] extend it to $\mathsf{OR}(K \otimes M) \geq \mathsf{rk}_\vee(K) \cdot |M|/kl$ for any square matrix K. To the best of our knowledge, the current literature has no stronger lower bounds on the OR-complexity of Kronecker products; our Theorem 13 comes logarithmically close to the desired bound. For SUM-complexity, the state of the art and our contribution are analogous to the OR-case. The related notion of a fractional biclique cover has previously appeared, e.g., in the papers of Watts [42] and Jukna and Kulikov [15].

Also related to our work is the study of the size of smallest biclique coverings, under the name of the bipartite dimension of a graph (as opposed to the cost of such coverings and the OR_2-complexity; see Section 3). This quantity corresponds to the Boolean rank of a matrix and is known to be PSPACE-hard to compute [9] and NP-hard to approximate to within a factor of $n^{1-\varepsilon}$ [3]. Finally, we note that results on OR_2-complexity have corollaries for **descriptional complexity** of regular languages. Indeed, take a language where all words have length two, $L \subseteq \Sigma \cdot \Delta$, with $\Sigma = \{a_1, \ldots, a_m\}$ and $\Delta = \{a_1, \ldots, a_n\}$. Let M^L be its characteristic $m \times n$ matrix: $M^L_{i,j} = 1$ iff $a_i \cdot a_j \in L$. Then $\mathsf{OR}_2(M^L)$ coincides with the alphabetic length of the shortest regular expression for L; for example, it follows from Corollary 5 that the optimal regular expression for the language $L_n = \{a_i a_j \mid 1 \leq i < j \leq n\}$ has $n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$ occurrences of letters ($\Sigma = \Delta = \{a_1, \ldots, a_n\}$). The values of $\mathsf{OR}(M^L)$ and $\mathsf{OR}_2(M^L)$ are also related to the size of the smallest nondeterministic finite automata accepting L; see [12] and Appendix for details.

---

[2] Recall that the $\mathsf{SUM}(\cdot)$ measure corresponds to *unambiguous* rectifier networks, in which every input-output pair is connected by at most one path; or, equivalently, to arithmetic circuits over nonnegative integers with addition (SUM) gates. For any matrix M, $\mathsf{OR}(M) \leq \mathsf{SUM}(M)$ and $\mathsf{OR}_2(M) \leq \mathsf{SUM}_2(M)$.

[3] In some contexts, the terms "direct sum theorem" and "direct product theorem" have slightly different meanings [36], but in the current context we do not distinguish between them.



$$B = \begin{pmatrix} 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 1&1&1&1&1&1&1&1\\ 0&0&0&0&1&1&1&1\\ 0&0&0&0&1&1&1&1\\ 0&0&0&0&1&1&1&1\\ 0&0&0&0&1&1&1&1 \end{pmatrix}$$

(a) Rectifier network of depth 3 **(b)** Matrix B **(c)** Rectifier network of depth 2

Figure 1 Illustrations for Example 1

3 Rectifier networks and coverings

Rectifier networks

Define a *rectifier network* with m inputs and n outputs as a 4-tuple $\mathcal{N} = (V, E, \mathsf{in}, \mathsf{out})$, where V is a set of vertices, $E \subseteq V^2$ a set of edges such that the directed graph $G_\mathcal{N} = (V, E)$ is acyclic, and $\mathsf{in}\colon \{1, \ldots, n\} \to V$ and $\mathsf{out}\colon \{1, \ldots, m\} \to V$ are injective functions whose images contain only sources (and, respectively, only sinks) of $G_\mathcal{N}$. The network $\mathcal{N}$ is said to have *size* $|E|$.

A rectifier network $\mathcal{N}$ *expresses* a Boolean $m \times n$ matrix $M = M(\mathcal{N})$ such that $M_{ij} = 1$ if $G_\mathcal{N}$ contains a directed path from $\mathsf{in}(j)$ to $\mathsf{out}(i)$ and $M_{ij} = 0$ otherwise. A rectifier network $\mathcal{N}$ is said to have *depth* d if all maximal paths in $G_\mathcal{N}$ have exactly d edges. Given a Boolean matrix $A \in \{0, 1\}^{m\times n}$, let $\mathsf{OR}_2(A)$ denote the smallest size of a depth-2 rectifier network that expresses A and let $\mathsf{OR}(A)$ denote the smallest size of any rectifier network that expresses A.

This notation is justified by the following observation. A rectifier network $\mathcal{N}$ may be viewed as a circuit: its Boolean inputs are located at the vertices $\mathsf{in}(\{1, \ldots, n\})$, and gates at all other vertices compute the disjunction (Boolean OR) of their inputs. From this point of view, the circuit computes a linear operator over the monoid $(\{0, 1\}, \mathrm{OR})$, and the matrix of this linear operator is exactly the Boolean matrix expressed by the rectifier network $\mathcal{N}$.

▶ **Example 1.** A depth-3 rectifier network is shown in Figure 1a. It expresses the matrix B in Figure 1b, showing that $\mathsf{OR}_3(B) \le 19$. In fact, this network is optimal and $\mathsf{OR}_3(B) = 19$; see Appendix for details. At the same time, $\mathsf{OR}_2(B) = 20$: the upper bound is achieved by the network in Figure 1c, and the lower bound is due to Jukna and Sergeev [16, Theorem 3.18].

Coverings of Boolean matrices

Let us describe an alternative way of defining the function $\mathsf{OR}_2(\cdot)$. Given a Boolean matrix A, a *rectangle* (or a 1-rectangle) is a pair (R, C), where $R \subseteq \{1, \ldots, m\}$ and $C \subseteq \{1, \ldots, n\}$, such that for all $(i, j) \in R \times C$ we have $A_{ij} = 1$. A rectangle (R, C) is said to *cover* all pairs $(i, j) \in R \times C$. The *cost* of a rectangle (R, C) is defined as $|R| + |C|$.

Suppose a matrix A is fixed; then a collection of rectangles is called a *covering* of A if for every $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$ there exists a rectangle in the collection that covers (i, j). The *cost* of a collection is the sum of costs of all its rectangles.

Given a Boolean matrix $A \in \{0, 1\}^{m\times n}$, the *cost* of A is defined as the smallest cost of a covering of A. It is not difficult to show that the cost of A equals $\mathsf{OR}_2(A)$ as defined above.

Similarly, we can think of minimizing the *size* of a covering, i.e., the number of rectangles in a collection instead of their total cost. The smallest size of a covering of A is called the OR-*rank* (or the *Boolean rank*) of A, denoted $\mathsf{rk}_\vee A$.

$$\sum_{S\in\mathcal{F}} w(S)\, x_S \to \min$$
$$x_S \in \{0,1\} \text{ for all } S \in \mathcal{F}$$
$$\sum_{\substack{S\in\mathcal{F}:\\ u\in S}} x_S \geq 1 \text{ for all } u \in U$$

(a) Integer program

$$\sum_{S\in\mathcal{F}} w(S)\, x_S \to \min$$
$$0 \leq x_S \leq 1 \text{ for all } S \in \mathcal{F}$$
$$\sum_{\substack{S\in\mathcal{F}:\\ u\in S}} x_S \geq 1 \text{ for all } u \in U$$

(b) Linear relaxation

$$\sum_{u\in U} y_u \to \max$$
$$y_u \geq 0 \text{ for all } u \in U$$
$$\sum_{u\in S} y_u \leq w(S) \text{ for all } S \in \mathcal{F}$$

(c) Dual of the linear relaxation

Figure 2 Integer and linear programs for the set cover problem

4 Fractional and greedy coverings

In the rest of the paper we interpret the covering problems for Boolean matrices as special cases of the general set cover problem. In this section we recall this general setting and present two main techniques that we apply: linear programming duality and greedy heuristics.

An instance of the *(weighted) set cover* problem consists of a set U, a family of its subsets, $\mathcal{F} \subseteq 2^U$, and a weight function, which is a mapping $w\colon \mathcal{F} \to \mathbb{N}$. Every set $S \in \mathcal{F}$ is said to *cover* all elements $s \in S \subseteq U$. The goal is to find a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ that is a *covering* (i.e., it covers all elements from U: $\bigcup_{S\in\mathcal{F}'} S = U$) and has the smallest possible total weight (i.e., it minimizes the functional $\sum_{S\in\mathcal{F}'} w(S)$ amongst all coverings). In the *unweighted* version of the problem, $w(S) = 1$ for all $S \in \mathcal{F}$, so the total weight of a covering is just its *size* (number of elements in $\mathcal{F}'$). In both versions, $\mathcal{F}$ is usually assumed to be a feasible solution, which means that every $s \in U$ belongs to at least one set from $\mathcal{F}$: that is, $\bigcup_{S\in\mathcal{F}} S = U$.

It is instructive, throughout this section, to have particular instances of the set cover problem in mind, namely those of covering Boolean matrices with rectangles as in Section 3. In the following sections, we refer to them as *weighted* and *unweighted set covering formulations*; their optimal solutions correspond to the values of $\mathsf{OR}_2(A)$ and $\mathsf{rk}_\vee A$ respectively.

Fractional coverings

The set cover problem can easily be recast as an integer program: see Figure 2a. For each $S \in \mathcal{F}$, this program has an integer variable $x_S \in \{0,1\}$: the interpretation is that $x_S = 1$ if and only if $S \in \mathcal{F}'$, and the constraints require that every element is covered. *Feasible* solutions are in a natural one-to-one correspondence with coverings of U, and the optimal value in the program is the smallest weight of a covering.

The *linear programming relaxation* of this integer program is obtained by interpreting variables x_S over reals: see Figure 2b. Now $0 \leq x_S \leq 1$ for each $S \in \mathcal{F}$. Feasible solutions to this program are called *fractional coverings*. Suppose the optimal cost in the original set cover problem is τ. Then the integer program in Figure 2a has optimal value τ, and its relaxation in Figure 2b optimal value $\tau^* \leq \tau$.

Finally, define the *dual* of this linear program: this is also a linear program, and it has a (real) variable y_u for each element $u \in U$; see Figure 2c. This is a maximization problem, and its optimal value coincides with τ^* by the strong duality theorem.

The following lemma summarizes the properties of these programs needed for the sequel.

▶ **Lemma 2.** *If $(y_u)_{u\in U}$ is a feasible solution to the dual, then $\sum_{u\in U} y_u \leq \tau^* \leq \tau$. There exists a feasible solution to the dual, $(y^*_u)_{u\in U}$, such that $\sum_{u\in U} y^*_u = \tau^*$.*

The proof can be found in, e.g., [17]. We use the first part of Lemma 2 in Section 5 to obtain a lower bound on τ and the second part in Section 7 to associate "weights" with 1-elements in the matrix.

Greedy coverings

The greedy heuristic for the unweighted set cover problem works as follows. It maintains the set of uncovered elements, initially U, and iteratively adds to $\mathcal{F}'$ (which is initially empty) a set $S \in \mathcal{F}$ which covers the largest number of yet-uncovered elements. Any covering obtained by this (nondeterministic) procedure is called a *greedy covering*. (There is a natural extension to the weighted version as well.)

A standard analysis of the greedy heuristic is performed in the framework of approximation algorithms: the size of a greedy covering is at most $O(\log |U|)$ times larger than that of the optimal covering [4, 24]. But for our purposes a different upper bound will be more convenient: an "absolute" upper bound in terms of the "density" of the instance. Such a bound is given by the following result, which is substantially less well-known:

▶ **Lemma 3** (greedy covering lemma). *Suppose every element $s \in U$ is contained in at least $\gamma|\mathcal{F}|$ sets from $\mathcal{F}$, where $0 < \gamma \le 1$. Then the size of any greedy covering does not exceed*

$$\left\lceil \frac{1}{\gamma} \ln^{+}(\gamma|U|) \right\rceil + \frac{1}{\gamma},$$

where $\ln^{+}(x) = \max(0, \ln x)$ and $\ln x$ is the natural logarithm.

Several versions of the lemma can be found in the literature. It was proved for the first time in 1972 by Sapozhenko [34] and appears in later textbooks [40, Lemma 9 in Section 3, pp. 136–137], [41, pp. 134–135]. A slightly different form, attributed to Stein [38] and Lovász [23], was independently obtained later and is sometimes known as the Lovász–Stein theorem; yet another proof is due to Karpinski and Zelikovsky [18]. Recent treatments with applications and more detailed discussion can be found in Deng et al. [5] and in Jukna's textbook [14, pp. 34–37].

Since the upper bound of Lemma 3 is hardly a standard tool in theoretical computer science as of now, a remark on the proof is in order. A standalone proof goes via the following fact: on each step of the greedy algorithm the number of yet-uncovered elements shrinks by a constant factor, determined by the density parameter γ and the size of the instance. Alternatively, one can use the result due to Lovász [23] that the size of any greedy covering is within a factor of $1 + \log |U|$ from the optimal *fractional* covering. Since assigning the value $(\min_{s \in U} |\{S \in \mathcal{F} \colon s \in S\}|)^{-1} = 1/\gamma|U|$ to all x_S, $S \in \mathcal{F}$, in the linear program in Figure 2b leads to a feasible solution, an upper bound of $(1/\gamma) \cdot (1 + \log |U|)$ follows.

We use Lemma 3 in Section 6 to obtain an upper bound on the OR_2-complexity of Kneser-Sierpiński matrices. We remark that instead of greedy coverings one can use random coverings to essentially the same effect (cf. Deng et al. [5]).

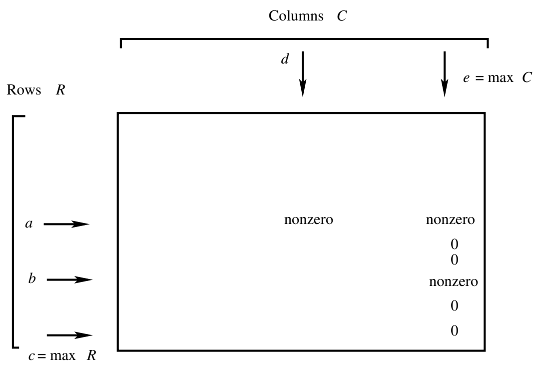
5 Lower bound for the full triangular matrices

Define the $n \times n$ *full triangular matrix* $T_n = (t_{ij})_{0 \le i,j < n}$ by $t_{ij} = 1$ if $i < j$ and $t_{ij} = 0$ otherwise. This matrix T_n is the adjacency matrix of the Hasse diagram of the strict linear order $0 < 1 < \cdots < n-1$; it has 1s everywhere above the main diagonal and 0s on the diagonal and below. In this section, we study the smallest size of depth-2 rectifier networks that express T_n.

Define $s(n) = n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$ for $n \ge 1$. Note that $s(n)$ is the so-called binary entropy function, sequence A003314 in Sloane's *Encyclopedia of Integer Sequences* [37]. Its properties were studied previously by Morris [29] because of its connection with mergesort.

$i\backslash j$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	2	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	2	1	0	1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	2	1	0	1	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	2	1	0	1	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	2	1	0	1	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	2	1	0	1	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	2	1	0	1	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0	1	0
8	0	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0	1
9	0	0	0	0	0	0	0	0	0	0	2	1	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	2	1	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) Portion of the matrix M



(b) Definition of a, b, c, d, e

Figure 3 Illustrations for the proof of Theorem 4

▶ **Theorem 4.** *All fractional coverings of T_n have cost of at least $s(n)$.*

▶ **Corollary 5.** $\mathsf{OR}_2(T_n) = s(n)$.

Note that the equality of Corollary 5 gives the exact value of $\mathsf{OR}_2(T_n)$. The upper bound is an easy divide-and-conquer argument (reproduced in Appendix for completeness), and the main challenge is to obtain the lower bound.

Consider the weighted set covering formulation for T_n, where the optimal value is $\mathsf{OR}_2(T_n)$ as discussed in Section 4. By Lemma 2, it suffices to find a feasible solution to the dual linear program with the value $s(n)$. Our feasible solution is given by a certain infinite diagonal matrix M, with rows and columns indexed by the natural numbers, defined as follows:

$$M_{i,j} = \begin{cases} 2, & \text{if } j - i = 1; \\ 1, & \text{if } j - i = 2^q \text{ for some } q \geq 1; \\ 0, & \text{otherwise.} \end{cases}$$

The first 17 rows and columns of M are displayed in Figure 3a. Notice that each row is a shift, by 1, of the preceding row.

▶ **Lemma 6.** *The sum of the elements of $M^{(n)}$, the $n \times n$ upper left submatrix of M, is equal to $s(n)$.*

▶ **Lemma 7.** *$y_{i,j} = M_{i,j}$ for $0 \leq i < j < n$ is a feasible solution to the dual program.*

Proof of Lemma 6. $M^{(n+1)}$ is obtained from $M^{(n)}$ by concatenating a row of 0's on the bottom, and a column that contains a single 2 and 1's corresponding to the powers of 2 that are $\leq n$. In other words, $s(n+1) = s(n) + \lfloor \log_2 n \rfloor + 2$. The result now follows by an easy induction. ◀

Proof of Lemma 7. To prove feasibility, we need to see that for each pair of nonempty sets $R, C \subseteq \{0, 1, \ldots, n-1\}$ with $\max R < \min C$—only such pairs (R, C) are rectangles of T_n—we have

$$\sum_{\substack{i \in R \\ j \in C}} M_{i,j} \leq |R| + |C|. \tag{1}$$

Here R corresponds to a choice of rows of M and C to a choice of columns.

Suppose there exists a counterexample to (1). Among all counterexamples to (1), consider one with the smallest possible value of $|R|+|C|$. If $|R|=1$ then since at most one entry in each row is 2 and all others are either 0 or 1, we clearly have $\sum_{\substack{i\in R\\ j\in C}} M_{i,j} \leq |R|+|C| = |C|+1$. Hence $|R|\geq 2$. The same argument applies if $|C|=1$. Thus the minimal counterexample to (1) has at least two rows and columns.

We now observe that the row sum of each row in our counterexample is at least 2. For if it is 0 or 1 we could omit that row, and (1) would still be violated. The same argument applies to the column sums. We now prove

▶ Claim. Suppose there are at least two nonzero elements in the submatrix of M formed by rows $0,1,\ldots,b$ and column e of M. Then $e \leq 2b$.

Proof. The nonzero elements in column e occur precisely in the rows numbered $e-1, e-2,\ldots,e-2^i$ where i is the largest integer with $e-2^i \geq 0$. So if there are nonzero elements in rows $0,1,\ldots,b$, these would be given by $e-2^i$ and $e-2^{i-1}$. So $e-2^{i-1}\leq b$. It now follows that $e \leq b+2^{i-1} = b+\frac{1}{2}\cdot 2^i \leq b+\frac{1}{2}e$ (since $e\geq 2^i$), and so $e\leq 2b$. This concludes the proof of the claim. ◀

Now let us assume that our minimal counterexample has $c=\max R$. Let $e=\max C$. Since column e has 2 nonzero elements, by the Claim above we know $e\leq 2c$. Now let b be the largest element $\leq c$ in R for which there is a nonzero element in column e; this must exist since column e has at least two nonzero elements. Let a be any row $< b$ in R with a nonzero element in column e. Again, this must exist since column e has at least two nonzero elements. Finally, let d be any column $< e$ in C with a nonzero element in row a. This must exist because every row in R has at least two nonzero elements. We claim $d\leq c$.

To see this, note that $b = e-2^j \leq c$ for some $j\geq 0$. (In fact, $j=\lceil \log_2(e-c)\rceil$.) Then we must have $a=e-2^k\geq 0$ where $k\geq j+1$. Then $d-a=2^\ell$ for some ℓ. So $d-a = d-(e-2^k) = 2^\ell$ and hence $d=e+2^\ell-2^k$. Since $d<e$ we have $\ell<k$. So $d\leq e+2^{k-1}-2^k = e-2^{k-1}\leq e-2^j = b\leq c$. This is illustrated in Figure 3b.

Now $\max R < \min C$, but $d\leq c$ while $d\in C$ and $c\in R$, a contradiction. Hence there are no minimal counterexamples and no counterexamples at all. Thus (1) holds. It follows that M represents a feasible solution. This concludes the proof of Lemma 7. ◀

Let us complete the proof of Theorem 4. Apply the first part of Lemma 2 to the weighted set covering formulation of the problem and take the solution $y_{i,j} = M_{i,j}$, $0\leq i<j<n$, as described above. This solution has value $s(n)$ by Lemma 6 and is feasible by Lemma 7. Hence, all fractional coverings have cost at least $s(n)$.

6 Upper bound for Kneser-Sierpiński matrices

Suppose $n=2^k$. A *Kneser-Sierpiński matrix* (or a *disjointness matrix*) of size $2^k\times 2^k$ is the matrix D_n defined as follows. Rows and columns of the matrix are indexed from 0 to 2^k-1. The matrix has a 1 at all positions (i,j) such that i and j have no common 1 in their binary expansion; all other elements of the matrix are 0.

Note that if we identify each number from $\{0,\ldots,n-1\}$ with a subset of $\{1,\ldots,k\}$ in the natural way, then D_n is naturally associated with a Boolean function that maps a pair of subsets of $\{1,\ldots,k\}$ to 1 if they are disjoint, and to 0 if they have an element in common. An alternative way to define D_n is by a recurrence $D_{2n} = \left(\begin{smallmatrix} D_n & D_n \\ D_n & 0\end{smallmatrix}\right)$ for $n\geq 1$; $D_1=(1)$;

here subsets of $\{1,\ldots,k\}$ are ordered lexicographically. Using the antilexicographic order for rows and the lexicographic order for columns would lead to a lower triangular matrix.

What is the size of smallest depth-2 rectifier networks that express Kneser-Sierpiński matrices? Jukna and Sergeev [16, Lemma 4.2] prove that

$$n^{\frac{1}{2}\log 5}/\operatorname{polylog}(n) \le \mathsf{OR}_2(D_n) \le n^{\log(1+\sqrt{2})} \cdot \operatorname{polylog}(n), \tag{2}$$

and in this section, we prove the following result:

▶ **Theorem 8.** $\mathsf{OR}_2(D_n) \le n^{\log(9/4)} \cdot \operatorname{polylog}(n)$.

Note that $\frac{1}{2}\log 5 \approx 1.16096$, $\log(9/4) \approx 1.16993$, and $\log(1+\sqrt{2}) \approx 1.27$.

Suppose $n = 2^k$ as above, and let $D^{x,y}_{[k]}$ be the submatrix of D_n whose rows and columns correspond to x-sized and y-sized subsets of $\{1,\ldots,k\}$, respectively. This matrix $D^{x,y}_{[k]}$ has size $\binom{k}{x} \times \binom{k}{y}$. If $x = y$, then $D^{x,x}_{[k]}$ is the adjacency matrix of the Kneser graph [25].

For $0 \le y \le x \le k$, write $z = (k-x-y)/2$ and $f(x,y) = \binom{k}{x,z,k-x-z}/\binom{2z}{z}$.[4] Jukna and Sergeev [16, Lemma 4.2] show that all coverings of $D^{x,x}_{[k]}$ have cost at least $f(x,x)/\operatorname{poly}(k)$, and this gives the lower bound in equation (2): taking $x = 0.4k$ brings $f(x,x)$ to its maximum of $n^{\frac{1}{2}\log 5}$, if we disregard factors polylogarithmic in $n = 2^k$. Our Theorem 8 follows from Lemmas 9 and 11 below.

▶ **Lemma 9.** *There exists a covering of $D^{x,y}_{[k]}$ with cost at most $f(x,y) \cdot \operatorname{poly}(k)$.*

Proof. Consider $\mathcal{F}$, the family of all *ordered bipartitions* of $\{1,\ldots,k\}$ into sets of size $x+z$ and $y+z$, where $z = (k-x-y)/2$. Technically, an ordered bipartition is simply a subset of $\{1,\ldots,k\}$, but it is more instructive to view it as an ordered pair: this subset and its complement. Every such bipartition, $(S,\overline{S})$, corresponds to a (maximal) rectangle in $D^{x,y}_{[k]}$; elements of $D^{x,y}_{[k]}$ covered by the rectangle are pairs (X,Y) of disjoint sets that *respect* the bipartition: $X \subseteq S$ and $Y \subseteq \overline{S}$.

Use the greedy covering lemma (Lemma 3) for the unweighted set covering formulation with $\mathcal{F}$. There are $\binom{k}{x+z}$ bipartitions in this family, and every pair of disjoint sets (X,Y) of size x and y respects $\binom{2z}{z}$ of them, so $\gamma = \binom{2z}{z}/\binom{k}{x+z}$ and any greedy covering will contain at most N sets, where

$$N = \frac{\binom{k}{x+z}}{\binom{2z}{z}} \cdot (1+\ln(4^k)) + 1 = \frac{\binom{k}{x+z}}{\binom{2z}{z}} \cdot \operatorname{poly}(k).$$

For every bipartition in the covering, the corresponding 1-rectangle in $D^{x,y}_{[k]}$ will include $\binom{x+z}{z}$ rows and $\binom{y+z}{z}$ columns; its cost will be at most $2\binom{x+z}{z}$ as $y \le x$. So the total cost of the covering will not exceed

$$\binom{x+z}{z} \cdot 2N = \frac{2\binom{k}{x+z}\binom{x+z}{z}\cdot\operatorname{poly}(k)}{\binom{2z}{z}} = \frac{\binom{k}{x,z,k-x-z}\cdot\operatorname{poly}(k)}{\binom{2z}{z}} = f(x,y)\cdot\operatorname{poly}(k). \qquad ◀$$

▶ **Corollary 10.** *Suppose $0 \le m \le k/2$ and let $D^m_k = D^{m,m}_{[k]}$ be the adjacency matrix of the (bipartite) Kneser graph: vertices in each part are size-m subsets of $\{1,\ldots,k\}$, and two vertices from different parts are adjacent if and only if the subsets are disjoint. Then $d(m,k)/\operatorname{poly}(k) \le \mathsf{OR}_2(D^m_k) \le d(m,k)\cdot\operatorname{poly}(k)$ where $d(m,k) = \binom{k}{m,k/2-m,k/2}/\binom{k-2m}{k/2-m}$.*

[4] We use the standard notation for multinomial coefficients: $\binom{k}{a,b,c} = \frac{k!}{a!\,b!\,c!}$ provided that $a+b+c=k$.

▶ **Lemma 11.** *If* $0 \le y \le x \le k$*, then* $f(x,y) \le 2^{k\log(9/4)} \cdot \mathrm{poly}(k)$*, and there exists a pair* (x^*, y^*) *such that* $f(x^*, y^*) \ge 2^{k\log(9/4)} / \mathrm{poly}(k)$*.*

Proof. As above, let $2z = k - (x+y)$. Denote $\alpha = z/k$ and recall that the values of the binomial coefficients may be estimated with the help of the binary entropy function (not to be confused with $s(n)$ from Section 5, also known under this name): $\binom{k}{\lambda k} \sim 2^{H(\lambda)k + O(\log k)}$ as $k \to \infty$, where $H(\lambda) = -\lambda \log \lambda - (1-\lambda)\log(1-\lambda)$. This formula follows from Stirling's approximation for the factorial [7, Chapter 9 and Solution to Exercise 9.42]. Now

$$f(x,y) = \frac{\binom{k}{z}\binom{k-z}{x}}{\binom{2z}{z}} \le \frac{\binom{k}{z}\binom{k-z}{(k-z)/2}}{\binom{2z}{z}} = \frac{2^{kH(\alpha)}2^{(1-\alpha)kH(1/2)}}{2^{2\alpha k H(1/2)}} \cdot \mathrm{poly}(k) = 2^{(H(\alpha)+1-3\alpha)k} \cdot \mathrm{poly}(k)$$

as $H(1/2) = 1$. Simple calculations show that for $0 < \alpha < 1/2$ the inequality $H(\alpha) + 1 - 3\alpha \le H(1/9) + 1 - 3 \cdot 1/9 = \log(9/4)$ holds. This corresponds to $x = 4/9 \cdot k$ and $y = 3/9 \cdot k$. ◀

To complete the proof of Theorem 8, it remains to note that a union of coverings of matrices $D^{x,y}_{[k]}$ for all pairs x, y with $0 \le x, y \le k$ constitutes a covering of D_n. For $0 \le y \le x \le k$, the coverings are constructed by Lemma 9, and for $x \le y$ the construction just swaps the roles of x and y. Since there are only $(k+1)^2 = \mathrm{polylog}(n)$ pairs x, y in total, the desired follows from Lemma 11.

▶ Remark 12. Although Theorem 8 leaves a gap between the bounds on $\mathsf{OR}_2(D_n)$, the greedy strategy is, in fact, optimal: For each $D^{x,y}_{[k]}$, it suffices to use bipartitions into sets of size ℓ and $k - \ell$, for some $\ell = \ell(k; x, y)$. (See Appendix for more details.) Our choice of ℓ in Lemma 9 is $\ell = x + (k-x-y)/2$, and the optimal choice, $\ell = \ell^*(k; x, y)$, will deliver a tight upper bound on $\mathsf{OR}_2(D_n)$. Numerical experiments seem to indicate that the actual value of $\mathsf{OR}_2(D_n)$ is within a $\mathrm{polylog}(n)$ factor from $n^{\frac{1}{2}\log 5}$, but no formal proof is known to us.

7 Lower bound for Kronecker products

Given two matrices $K \in \{0,1\}^{m_1 \times n_1}$ and $M \in \{0,1\}^{m_2 \times n_2}$, their *Kronecker* (or *tensor*) *product* is the Boolean matrix $K \otimes M$ of size $(m_1 \cdot m_2) \times (n_1 \cdot n_2)$ defined as follows. Its rows are indexed by pairs (i_1, i_2) and its columns by pairs (j_1, j_2) where $1 \le i_s \le m_s$ and $1 \le j_s \le n_s$ for $s = 1, 2$. The entry of $K \otimes M$ at position $((i_1, i_2), (j_1, j_2))$ is defined as $K_{i_1,j_1} \cdot M_{i_2,j_2}$.

In this section we prove a lower bound on the $\mathsf{OR}(\cdot)$-measure of Kronecker products. Recall that the Boolean rank $\mathsf{rk}_\vee(K)$ is the optimal value of the unweighted set covering formulation (as in Figure 2a) where the set of 1-entries in the matrix K is covered by all-1 rectangles. In the linear relaxation of this problem (as in Figure 2b), the goal is to assign weights $w(R) \in [0,1]$ to each 1-rectangle R such that $\sum_{(i,j)\in R} w(R) \ge 1$ for each 1-entry (i,j) of K, minimizing $\sum w(R)$. Let *the fractional rank* $\mathsf{rk}^*_\vee(K)$ be the optimal value of this linear relaxation. The integrality gap result for the set cover problem [23] and the duality theorem imply that $\mathsf{rk}_\vee(K)/(1 + \log m_1 n_1) \le \mathsf{rk}^*_\vee(K) \le \mathsf{rk}_\vee(K)$. In the graph-theoretic language, the number $\mathsf{rk}^*_\vee(K)$ is the *fractional biclique cover number*, denoted by $bc^*(G)$ where K is the adjacency matrix of the (bipartite) graph G. Fractional rank is known to be bounded from below by the fooling set number, see Watts [42, Theorem 2.2].

▶ **Theorem 13.** *For any pair* K*,* M *of Boolean matrices,* $\mathsf{OR}(K \otimes M) \ge \mathsf{rk}^*_\vee(K) \cdot \mathsf{OR}(M)$*.*

Proof. First consider the unweighted set covering formulation for K, where the optimal value is $\mathsf{rk}_\vee(K)$ as discussed in Section 4, and take its linear relaxation, with the optimal value $\mathsf{rk}^*_\vee(K)$. By Lemma 2, there is an assignment of weights to 1-elements of this matrix, $w(i,j) \in [0,1]$ for all (i,j) with $K_{i,j} = 1$, such that the following two conditions are satisfied (see Figure 2c). First, for each 1-rectangle $R \times C$ of K, the sum $\sum_{(i,j)\in R\times C} w(i,j)$ is at most 1. Second, $\sum_{(i,j):K_{i,j}=1} w(i,j) = \mathsf{rk}^*_\vee(K)$.

Now let $\mathcal{N} = (V, E, \mathsf{in}, \mathsf{out})$ be a rectifier network of size $\mathsf{OR}(K \otimes M)$ that expresses $Q = K \otimes M$, where K and M have size as above. For an edge $e \in E$, let $\mathrm{To}(e) \subseteq \{1,\dots,m_1\} \times \{1,\dots,m_2\}$ be the set of row indices (i_1,i_2) of Q such that the node $\mathsf{out}(i_1,i_2)$ is reachable from the target of e. Similarly, let $\mathrm{From}(e) \subseteq \{1,\dots,n_1\} \times \{1,\dots,n_2\}$ be the set of column indices (j_1,j_2) of Q such that the source of e is reachable from $\mathsf{in}((j_1,j_2))$. Then $R(e) = (\mathrm{To}(e), \mathrm{From}(e))$ is a rectangle of Q. Moreover, define $\pi_s((i_1,i_2),(j_1,j_2)) = (i_s,j_s)$ for $s = 1,2$ and $\pi_s(R) = \{\pi_s(r,c) \colon (r,c) \in R\}$. Then $\pi_1(R(e))$ and $\pi_2(R(e))$ are rectangles in K and M respectively.

We assign real weights based on w to each edge e of $\mathcal{N}$ by the following rule:

$$w'(e) = \sum_{(i,j)\in\pi_1(R(e))} w(i,j).$$

Since $\pi_1(R(e))$ is a rectangle of K, one of the constraints on w ensures that $w'(e) \le 1$ for each edge e of $\mathcal{N}$. Consequently, $\sum_{e\in E} w'(e) \le |E| = \mathsf{OR}(K \otimes M)$; furthermore, the following chain of inequalities holds:

$$\begin{aligned}\mathsf{OR}(K\otimes M) \ge \sum_{e\in E} w'(e) &= \sum_{e\in E} \sum_{(i_1,j_1)\in\pi_1(R(e))} w(i_1,j_1)\\ &= \sum_{(i_1,j_1):K_{i_1,j_1}=1} w(i_1,j_1)\cdot|\{e\in E : (i_1,j_1)\in\pi_1(R(e))\}|\\ &= \sum_{(i_1,j_1):K_{i_1,j_1}=1} w(i_1,j_1)\cdot|\{e\in E : i_1\in\pi_1(\mathrm{To}(e)), j_1\in\pi_1(\mathrm{From}(e))\}|. \qquad (3)\end{aligned}$$

Fix an arbitrary entry (i_1,j_1) of K with $K_{i_1,j_1} = 1$. Consider the subgraph $\mathcal{N}_{j_1\rightsquigarrow i_1}$ of $\mathcal{N}$ induced by the nodes that are reachable from some source of the form $\mathsf{in}(j_1,j_2)$ and from which a node of the form $\mathsf{out}(i_1,i_2)$ is reachable—in other words, take all nodes and edges on all paths from $\mathsf{in}(j_1,j_2)$ to $\mathsf{out}(i_1,i_2)$ for some i_2, j_2. Then, since $K_{i_1,j_1} = 1$, the node $\mathsf{out}(i_1,i_2)$ is reachable from $\mathsf{in}(j_1,j_2)$ in $\mathcal{N}_{j_1\rightsquigarrow i_1}$ if and only if $M_{i_2,j_2} = 1$. So the network $\mathcal{N}_{j_1\rightsquigarrow i_1}$ expresses M (with the mappings $\mathsf{in}'(j_2) = \mathsf{in}(j_1,j_2)$ and $\mathsf{out}'(i_2) = \mathsf{out}(i_1,i_2)$). Hence, the number of edges in $\mathcal{N}_{j_1\rightsquigarrow i_1}$ is at least $\mathsf{OR}(M)$. But by our definitions, the relations $i_1 \in \pi_1(\mathrm{To}(e))$ and $j_1 \in \pi_1(\mathrm{From}(e))$ hold together exactly for the edges e of $\mathcal{N}$ present in $\mathcal{N}_{j_1\rightsquigarrow i_1}$. Thus $|\{e \in E : i_1 \in \pi_1(\mathrm{To}(e)), j_1 \in \pi_1(\mathrm{From}(e))\}| \ge \mathsf{OR}(M)$ and we conclude from equation (3) that

$$\mathsf{OR}(K\otimes M) \ge \sum_{(i_1,j_1):K_{i_1,j_1}=1} w(i_1,j_1)\cdot\mathsf{OR}(M) = \mathsf{rk}^*_\vee(K)\cdot\mathsf{OR}(M). \qquad \blacktriangleleft$$

▶ Remark 14. Let $\mathsf{SUM}(K)$ be the smallest size of an *unambiguous* rectifier network that expresses K. A rectifier network is unambiguous if for all i, j it has at most one path from $\mathsf{in}(j)$ to $\mathsf{out}(i)$. Such networks are also known under the names of SUM-circuits [16] and cancellation-free circuits [2]. The same construction as above also proves the inequality $\mathsf{SUM}(K \otimes M) \ge \mathsf{rk}^*_\vee(K) \cdot \mathsf{SUM}(M)$.

▶ **Corollary 15.** *For any pair of matrices $K \in \{0,1\}^{m_1\times n_1}$ and $M \in \{0,1\}^{m_2\times n_2}$, and $\mathsf{L} \in \{\mathsf{OR}, \mathsf{SUM}\}$ it holds that $\mathsf{L}(K \otimes M) \ge \mathsf{rk}_\vee(K) \cdot \mathsf{L}(M)/(1 + \log m_1 n_1)$.*

Acknowledgements. We are grateful to Stasys Jukna, Alexander Kulikov, Igor Sergeev, and anonymous reviewers for comments and discussions.

References

1 A.E. Andreev. On the complexity of gradient gate circuits. *Discrete Mathematics and Applications*, 5(1):83–92, 1995.

2 Joan Boyar and Magnus Gausdal Find. Cancellation-free circuits in unbounded and bounded depth. *Theor. Comput. Sci.*, 590:17–26, 2015.

3 Parinya Chalermsook, Sandy Heydrich, Eugenia Holm, and Andreas Karrenbauer. Nearly tight approximability results for minimum biclique cover and partition. In *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, pages 235–246, 2014.

4 V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

5 Dameng Deng, P.C. Li, G.H.J. van Rees, and Yuan Zhang. The Stein-Lovasz theorem and its applications to some combinatorial arrays. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 77:17–31, 2011.

6 Magnus Find, Mika Göös, Matti Järvisalo, Petteri Kaski, Mikko Koivisto, and Janne H. Korhonen. Separating OR, SUM, and XOR circuits. *CoRR*, abs/1304.0513, 2013.

7 Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley, 1998. 2nd edition.

8 Gregor Gramlich and Georg Schnitger. Minimizing NFA's and regular expressions. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 399–411, 2005.

9 Hermann Gruber and Markus Holzer. Finding lower bounds for nondeterministic state complexity is hard. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(027), 2006.

10 G. Hansel. Nombre minimal de contacts de fermeture nécessaires pour réaliser une fonction booléenne symétrique de n variables. *C. R. Acad. Sc. Paris*, 258(25):6037–6040, 1964. In French.

11 J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

12 Szabolcs Iván, Ádám Dániel Lelkes, Judit Nagy-György, Balázs Szörényi, and György Turán. Biclique coverings, rectifier networks and the cost of ϵ-removal. In *Descriptional Complexity of Formal Systems - 16th International Workshop, DCFS 2014, Turku, Finland, August 5-8, 2014. Proceedings*, pages 174–185, 2014.

13 Tao Jiang, Edward McDowell, and Bala Ravikumar. The structure and complexity of minimal NFA's over a unary alphabet. *Int. J. Found. Comput. Sci.*, 2(2):163–182, 1991.

14 Stasys Jukna. *Extremal Combinatorics.* Springer-Verlag, 2011. 2nd edition.

15 Stasys Jukna and Alexander S. Kulikov. On covering graphs by complete bipartite subgraphs. *Discrete Mathematics*, 309(10):3399–3403, 2009.

16 Stasys Jukna and Igor Sergeev. Complexity of linear Boolean operators. *Foundations and Trends in Theoretical Computer Science*, 9(1):1–123, 2013. Available at `http://lovelace.thi.informatik.uni-frankfurt.de/~jukna/Knizka/linear.pdf`.

17 H. Karloff. *Linear Programming.* Birkhäuser, 2008. 2nd printing.

18 Marek Karpinski and Alexander Zelikovsky. Approximating dense cases of covering problems. In *Network design: connectivity and facilities location*, volume 40 of *DIMACS*, pages 169–178. AMS, 1998.

19 Gyula Katona and Endre Szemerédi. On a problem of graph theory. *Studia Scientiarum Mathematicarum Hungarica*, 2:23–28, 1967.

20 R.E. Krichevskii. A minimal monotone contact scheme for a Boolean function of n variables. In *Diskretnyj Analiz (Discrete Analysis)*, volume 5, pages 89–92. Institute for Mathematics in the Siberian Section of the Academy of Sciences, Novosibirsk, 1965. In Russian.

21 Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

22 Jonathan Lee and Jeffrey Shallit. Enumerating regular expressions and their languages. In *Implementation and Application of Automata, 9th International Conference, CIAA 2004, Kingston, Canada, July 22-24, 2004, Revised Selected Papers*, pages 2–22, 2004.

23 L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975.

24 L. Lovász. A kombinatorika minimax tételeiről. *Matematikai Lapok*, 26:209–264, 1976. In Hungarian.

25 László Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Comb. Theory, Ser. A*, 25(3):319–324, 1978.

26 S.A. Lozhkin. On minimal π-circuits for monotone symmetric functions with threshold 2. *Diskretnaya Matematika*, 17(4):108–110, 2005. In Russian. English translation in *Discrete Mathematics and Applications* 15(5) (2005), 475–477.

27 Kurt Mehlhorn. Some remarks on Boolean sums. *Acta Inf.*, 12:371–375, 1979.

28 Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory, College Park, Maryland, USA, October 25-27, 1972*, pages 125–129, 1972.

29 R. Morris. Some theorems on sorting. *SIAM J. Appl. Math.*, 17:1–6, 1969.

30 E. I. Nechiporuk. Rectifier networks. *Soviet Physics Doklady*, 8:5–7, March 1963.

31 E.I. Nechiporuk. On the topological principles of self-correction. *Problemy Kibernetiki*, 21:5–102, 1969. In Russian. English translation in: Systems Theory Res. 21 (1970), 1–99.

32 Nicholas Pippenger. On another Boolean matrix. *Theor. Comput. Sci.*, 11:49–56, 1980.

33 Jaikumar Radhakrishnan. Entropy and counting. In J.C. Misra, editor, *Computational Mathematics, Modelling and Algorithms*. Narosa Publishers, New Delhi, 2003. Available online at `http://www.tcs.tifr.res.in/~jaikumar/Papers/EntropyAndCounting.pdf`.

34 Alexander Sapozhenko. On the complexity of disjunctive normal forms obtained with a gradient algorithm. In *Diskretnyj Analiz (Discrete Analysis)*, volume 21, pages 62–71. Institute for Mathematics in the Siberian Section of the Academy of Sciences, Novosibirsk, 1972. In Russian.

35 S.N. Selezneva. Lower bound on the complexity of finding polynomials of Boolean functions in the class of circuits with separated variables. *Computational Mathematics and Modeling*, 24(1):146–152, 2013.

36 Alexander A. Sherstov. Strong direct product theorems for quantum communication and query complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012.

37 N. J. A. Sloane. On-line encyclopedia of integer sequences. Electronic resource at `http://oeis.org`.

38 S. K. Stein. Two combinatorial covering theorems. *J. Comb. Theory, Ser. A*, 16(3):391–397, 1974.

39 T.G. Tarján. Complexity of lattice-configurations. *Studia Scientiarum Mathematicarum Hungarica*, 10:203–211, 1975.

40 Yu.L. Vasilyev and V.V. Glagolev. Metrical properties of disjunctive normal forms. In S.V. Yablonsky and O.B. Lupanov, editors, *Discrete mathematics and mathematical questions of cybernetics*, pages 99–148. Nauka, Moscow, 1974. In Russian.

41 J.L. Wassiljew and W.W. Glagolew. Metrische Eigenschaften alternativer Normalformen. In S.W. Jablonski and O.B. Lupanow, editors, *Diskrete Mathematik und Mathematische Fragen der Kybernetik*, volume 71 of *Mathematische Reihe*, pages 100–144. Birkhäuser Basel, 1980. In German.

42 Valerie L. Watts. Fractional biclique covers and partitions of graphs. *Electr. J. Comb.*, 13(1), 2006.

43 Ingo Wegener. A new lower bound on the monotone network complexity of Boolean sums. *Acta Inf.*, 13:109–114, 1980.

44 David P. Williamson and David B. Shmoys. *The design of approximation algorithms.* Cambridge University Press, 2011.

A Depth-3 lower bound in Example 1

Consider the matrix $M_n = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \otimes J_n$ for some $n \geq 1$ where J_n is the $n \times n$ all-one matrix. Known bounds give $\mathrm{OR}(M_n) \geq 4n+1$ and this bound is indeed attainable. For $\mathrm{OR}_3(M_n)$, i.e. realization by some rectifier network of exact depth 3 we show $\mathrm{OR}_3(M_n) = 4n+3$ using the following lemma:

▶ **Lemma 16.** *Suppose M is a Boolean matrix and $\mathcal{N} = (V, E, \mathsf{in}, \mathsf{out})$ is a rectifier network realizing M of some depth d. Then there exists a rectifier network $\mathcal{N}' = (V, E', \mathsf{in}, \mathsf{out})$ with $|\mathcal{N}'| \leq |\mathcal{N}|$ having depth at most d satisfying the following conditions:*

i) whenever the i_1th and the i_2th row are the same in M, then the sets $\{v \in V : (v, \mathsf{out}(i_1)) \in E'\}$ and $\{v \in V : (v, \mathsf{out}(i_2)) \in E'\}$ coincide;

ii) dually, whenever the j_1th and the j_2th column of M are the same, then $\{v \in V : (\mathsf{in}(j_1), v) \in E'\} = \{v \in V : (\mathsf{in}(j_2), v) \in E'\}$.

Proof. Let $v = \mathsf{in}(j)$ be a source node and let X_j stand for the set $\{w \in V : (v, w) \in E\}$ of its neighbours. Since $\mathcal{N}$ realizes M, the set of target nodes $\mathsf{out}(i)$ which are reachable in $\mathcal{N}$ is exactly the image under out of those indices i for which $M_{i,j} = 1$. Now for each column index j let j' be the index for which the jth and the j'th column of M is the same, $|X_{j'}|$ is the smallest possible among these sets and j' is the smallest among these indices. Note that j' is always well-defined and whenever the j_1th and the j_2th column coincide, then $j'_1 = j'_2$.

Then, define $\mathcal{N}_0$ as$(V, E_0, \mathsf{in}, \mathsf{out})$ with $E_0 = E - \{(\mathsf{in}(j), v)\} \cup \{(\mathsf{in}(j), v) : v \in X_{j'}\}$. (That is, we reattach the edges coming out from sources to the neighbours of the representative source of their equivalence class.)

Then by the choice of the values j' (in particular, with $|X_{j'}|$ having been minimized) we have that i) is satisfied, $\mathcal{N}_0$ also realizes M, the depth is not increased (if $\mathcal{N}$ is strictly levelled) and $|\mathcal{N}'| \leq |\mathcal{N}|$. Applying the analogous transformation to the targets we get a network $\mathcal{N}'$ satisfying ii) as well. ◀

Thus we get that there exists a depth-3 network of minimal size realizing M_n such that

- each source $\mathsf{in}(i)$ for $i = 1, \ldots, n$ have the same set X_1 of neighbours;
- each source $\mathsf{in}(i)$ for $i = n+1, \ldots, 2n$ have the same set X_2 of neighbours;
- each target $\mathsf{out}(j)$ for $j = 1, \ldots, n$ have the same set Y_1 of neighbours and
- each target $\mathsf{out}(j)$ for $j = n+1, \ldots, 2n$ have the same set Y_2 of neighbours

since the corresponding rows and columns coincide. In this network there are $n(|X_1| + |X_2| + |Y_1| + |Y_2|)$ edges in total between the outermost layers (and some additional edges between the two middle layers. Clearly none of these sets can be empty (since all the rows and columns are nonzero), and if any of them is a non-singleton set, the size of the network is at least $5n > 4n + 3$. So in order to go below $5n$, $X_1 = \{x_1\}$, $X_2 = \{x_2\}$ etc. have to be singleton sets. Now since not all rows (columns, resp.) are equal, $x_1 \neq x_2$ and $y_1 \neq y_2$ has to hold, and there is only one choice (because the sets are singletons) to wire the two middle layers together, namely adding the edges (x_1, y_1), (x_1, y_2) and (x_2, y_2), giving $4n + 3$ edges in total as optimal value for depth $d = 3$.

Note that if the network is not required to be strictly levelled, we can merge x_1 with y_1 and x_2 with y_1 and add only the edge (x_1, x_2) reaching the optimal bound $4n + 1$.

B Upper bound in Corollary 5

Recall that a SUM-*circuit* for a matrix M is the same as an *unambiguous* rectifier network: it is a rectifier network that has at most one path between any input—output pair. The smallest size of an unambiguous rectifier network that expresses M is denoted by $\mathsf{SUM}(M)$; similarly, $\mathsf{SUM}_2(M)$ is the smallest size of an unambiguous rectifier network of depth 2 that expresses M. In the same way as rectifier networks of depth 2 correspond to rectangle coverings, *unambiguous* rectifier networks of depth 2 correspond to rectangle *partitions* (that is, coverings with no overlap between rectangles). If one views the matrices as adjacency matrices of bipartite graphs, then the measures $\mathsf{OR}_2(\cdot)$ and $\mathsf{SUM}_2(\cdot)$ correspond to minimal biclique coverings and minimal biclique partitions, respectively. Clearly, $\mathsf{OR}(M) \le \mathsf{SUM}(M)$ and $\mathsf{OR}_\mathsf{d}(M) \le \mathsf{SUM}_\mathsf{d}(M)$ for each depth d. Also, if $M = \left(\begin{smallmatrix} M_1 & M_2 \\ M_3 & M_4 \end{smallmatrix}\right)$, then $\mathsf{SUM}(M) \le \sum_{i=1}^{4} \mathsf{SUM}(M_i)$.

We show below that $\mathsf{SUM}_2(T_n) \le s(n) = n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$. Theorem 4 will then imply that $\mathsf{OR}_2(T_n) = \mathsf{SUM}_2(T_n) = s(n)$.

First, let J_n be the $n \times n$ all-1 matrix and $J_{m,k}$ the $m \times k$ all-1 matrix. Clearly, $\mathsf{SUM}_2(J_{m,k})$ is $m + k$. Second, observe that $T_{2n} = \left(\begin{smallmatrix} T_n & J_n \\ 0 & T_n \end{smallmatrix}\right)$ and $T_{2n+1} = \left(\begin{smallmatrix} T_n & J_{n,n+1} \\ 0 & T_{n+1} \end{smallmatrix}\right)$. It follows that $\mathsf{SUM}_2(T_{2n}) \le 2\mathsf{SUM}_2(T_n) + 2n$ and $\mathsf{SUM}_2(T_{2n+1}) \le \mathsf{SUM}_2(T_n) + \mathsf{SUM}_2(T_{n+1}) + 2n + 1$. This shows, by induction, that $\mathsf{SUM}_2(T_n) \le s(n)$, since the induction basis is easily checked.

C Optimality of the greedy strategy for Kneser-Sierpiński matrices

Although Theorem 8 leaves a gap between the bounds of $\Omega(n^{1.16})$ and $O(n^{1.17})$ on $\mathsf{OR}_2(D_n)$, the greedy strategy is, in fact, optimal. We first give a brief sketch of the argument, and then fill in all the details below.

Consider the linear relaxation of the set covering formulation for each $D^{x,y}_{[k]}$. Note that only maximal rectangles (i.e., those associated with bipartitions) can participate in optimal fractional coverings. In fact, for any $\ell \in [x, k-y]$ there exists a fractional covering $\eta(\ell)$ of $D^{x,y}_{[k]}$ which uses only bipartitions into sets of size ℓ and $k-\ell$ and for which all "covering" constraints in the LP are tight; it suffices to pick a single ℓ since this fractional covering $\eta(\ell)$ uses *all* such bipartitions with multiplicity $1/\binom{k-(x+y)}{\ell-x}$. Hence, the problem reduces to an unweighted set covering formulation, where the greedy heuristic achieves a value within a factor of $1 + \log\binom{k}{x}\binom{k}{y} \le 1 + 2k = \mathrm{polylog}(n)$ of the optimum.

In more detail, first consider an arbitrary weighted set cover problem: let $S_1, \dots, S_k \subseteq U$ be the sets, with $w_i > 0$ being the cost of S_i. Let $\mu = \min\{\frac{w_i}{|S_i|} : i = 1, \dots, k\}$ be the best cost/utility ratio offered by the sets. Then, in the dual formulation of its LP relaxation, if one assigns uniformly μ to each element $u \in U$ of the universe, then each set S_i gets $\mu \cdot |S_i| \le w_i$ total charge, hence this uniform distribution is a solution to the dual, hence $\mu \cdot |U|$ is a lower bound for the optimum of the primal problem by the weak duality theorem.

For the case of the weighted covering by rectangles, a rectangle of size $k \times m$ has cost $k + m$ and covers km elements, hence its offered ratio is $\frac{k+m}{km} = \frac{1}{k} + \frac{1}{m}$, i.e. it decreases strictly by increasing either k or m, thus the best ratios are always offered by maximal rectangles.

Now considering a rectangle R in a matrix $D^{x,y}_{[k]}$, formed by the rows $X_1, X_2, \dots, X_k$ and columns $Y_1, \dots, Y_m$ we have by definition that each X_i is disjoint from each Y_j, thus choosing $S = \bigcup_{i=1}^{k} X_i$ we have that R is a subrectangle of the rectangle corresponding to the bipartition $(S, \overline{S})$, yielding that only rectangles corresponding to bipartitions can be maximal.

On the other hand, any such rectangle is clearly maximal. Denoting $|S|$ by ℓ we get that the ratio offered by these rectangles is $\mu(k,x,y,\ell) = \frac{1}{\binom{\ell}{x}} + \frac{1}{\binom{k-\ell}{y}}$. Then setting $\ell^* = \ell^*(k,x,y) = \arg\min_\ell\{\mu(k,x,y,\ell) : x \le \ell, y \le k-\ell\}$ is the parameter of those rectangles offering the best possible ratio $\mu^* = \mu(k,x,y,\ell^*)$ for $D^{x,y}_{[k]}$. Thus, $\mu^* \cdot ||D^{x,y}_{[k]}|| = \frac{\binom{\ell^*}{x}+\binom{k-\ell^*}{y}}{\binom{\ell^*}{x}\binom{k-\ell^*}{y}}\left(\binom{k}{x}\binom{k-x}{y}\right)$ is a lower bound for the cost of the optimal solution.

Observe that this bound is indeed attainable by the greedy strategy, since each set (X,Y) with $|X| = x$ and $|Y| = y$, $X \cap Y = \emptyset$ is covered exactly by $\binom{k-x-y}{\ell^*-x}$ such rectangles (i.e. respects this number of such bipartitions), thus considering the fractional covering $\eta(\ell^*)$ which uses *all* such bipartitions with multiplicity $1/\binom{k-(x+y)}{\ell^*-x}$ we get a covering of $D^{x,y}_{[k]}$, with total cost $\frac{1}{\binom{k-(x+y)}{\ell^*-x}}\left(\binom{\ell^*}{x} + \binom{k-\ell^*}{y}\right)\binom{k}{\ell^*}$ (that is, multiplicity$\times$weight of a rectangle$\times$number of these rectangles). The last expression is the same as $\frac{\binom{\ell^*}{x}+\binom{k-\ell^*}{y}}{\binom{\ell^*}{x}\binom{k-\ell^*}{y}}\left(\binom{k}{x}\binom{k-x}{y}\right)$, since $\binom{k}{x}\binom{k-x}{y}\binom{k-(x+y)}{\ell^*-x} = \binom{k}{\ell^*}\binom{\ell^*}{x}\binom{k-\ell^*}{y}$: both of these products calculate the number of possibilities to choose an ℓ^*-element subset L of a k-element set K, and an x-element subset X of L as well as an y-element subset of $K - L$. The first formula achieves this by choosing X from K first, then Y from $K - X$, finally $L - X$ from $K - X - Y$, the second one by choosing L from K first, then X from L and finally Y from $K - L$. Thus, choosing all these bipartitions with this multiplicity provides an optimal solution.

Note that for any fixed ℓ, the weighted set covering problem using only the bipartitions $(S, \overline{S})$ with $|S| = \ell$ is a uniform-cost, i.e., an unweighted set covering problem. On such a problem the greedy heuristic achieves a value within a factor of $1 + \log\binom{k}{x}\binom{k}{y} \le 1 + 2k = \mathrm{polylog}(n)$ of the optimum in the linear relaxation. Therefore, it suffices to pick some ℓ and construct a greedy covering using bipartitions into sets of size ℓ and $k - \ell$. Our choice of ℓ in Lemma 9 is $\ell = x + (k - x - y)/2$, and the argument above shows that the optimal choice, $\ell = \ell^*(k; x, y)$ will deliver an upper bound on $\mathsf{OR}_2(D_n)$ that is tight up to a polylogarithmic factor, thus reducing the problem to a parametric optimization task.

D Application: size of regular expressions

A *regular expression* over Σ is a well-formed expression r consisting of the symbols

$$\epsilon, \emptyset, (,), +, *, \text{ and } a \in \Sigma,$$

with the usual semantics (e.g., as in [11]).

The *size* of a regular expression r can be specified in a number of different ways, but for our purposes, the easiest is the so-called *alphabetic length*, which is the number of symbols in r belonging to Σ [22]. For example, the alphabetic length of

$$r = a_0a_1 + a_2a_3 + (a_0 + a_1)(a_2 + a_3) \tag{4}$$

is 8.

Given a regular language L specified in some way (for example, as the language accepted by a finite automaton), it is, in general, quite difficult to determine the size of the shortest regular expression specifying L. In fact, this problem is PSPACE-hard [28, 13] and not even approximable within a factor of $o(n)$ [8] (unless P = PSPACE).

Extended example

In this subsection we examine a specific family of finite languages, namely

$$L_n = \sum_{0\leq i<j<n} a_i a_j,$$

over the alphabet $\Sigma_n = \{a_0, a_1, \ldots, a_{n-1}\}$ of size n, and we provide matching upper and lower bounds on for the size of the shortest regular expression for it. For example, for $n = 4$ this is the language

$$L_4 = \{a_0a_1, a_0a_2, a_0a_3, a_1a_2, a_1a_3, a_2a_3\}.$$

Evidently one can produce a regular expression for L_n of length $n(n-1)$ by listing the elements of L_n, but it is possible to do much better. For example, the regular expression given in (4) specifies L_4 with alphabetic length 8, as opposed to length 12 using the brute-force approach.

Our upper and lower bounds follow Corollary 5 in the main text. For the lower bound, we relate the alphabetic length of regular expressions to the cost of coverings of Boolean matrices; for the upper bound, we provide a direct proof to make the connection between regular expressions and coverings more transparent.

We first show how to construct a small regular expression for L_n through a simple divide-and-conquer strategy. We generalize L_n to $L_{A,B} = \bigcup_{A\leq i<j\leq B} a_i a_j$ so that $L_n = L_{0,n-1}$. Then our divide-and-conquer solution is given by

$$L_{A,B} = L_{A,C} \ \cup\ L_{C+1,B} \ \cup\ \{a_A + a_{A+1} + \cdots + a_C\} \cdot \{a_{C+1} + \cdots + a_B\},$$

where $C = \lfloor (A+B)/2 \rfloor$. The alphabetic length $t(n)$ of the regular expression so constructed satisfies the recurrence $t(1) = 0$ and $t(2n) = 2t(n)+2n$ and $t(2n+1) = t(n+1)+t(n)+2n+1$. Now an easy induction proves that in fact $t(n) = s(n)$, with $s(n) = n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$.

We now turn to the lower bound. Let r_n be a regular expression of shortest length for L_n for $n \geq 2$. Clearly we can assume that r_n contains no occurrence of the empty set symbol $\emptyset$. Since L_n is finite, we can also assume r_n contains no occurrence of $*$. So all the operators in r_n are either union or concatenation. Consider any instance of concatenation, say L_1L_2. Then if either L_1 or L_2 contains strings of two different lengths, the resulting concatenation would also, which is impossible since L_n contains only strings of length 2. So all strings on one side of any concatenation are of the same length. On the other hand, no strings can be of length 3 or more, and if one side contains only strings of length 0 (the empty string) we could simply omit the concatenation. So in fact we may assume, without loss of generality that any concatenation in r_n looks like $R \cdot C$, where both languages consist of subsets of Σ_n. Finally, every letter in C must be numbered higher than all those of R, for otherwise we would obtain a word not in L_n. This means that we can write r_n as

$$R_1 \cdot C_1 + R_2 \cdot C_2 + \cdots + R_t \cdot C_t \qquad (5)$$

where we have inserted dots to make the concatenation explicit. The alphabetic length of this expression is $\sum_{1\leq i\leq t}(|R_i| + |C_i|)$.

We now create an integer program to minimize this length. Define $\mathcal{I}_n = \{0, 1, \ldots, n-1\}$ and let $x_{R,C}$ for nonempty sets $R, C \subseteq \mathcal{I}_n$ be an indicator variable for the presence of the term $R \cdot C$ in the expression (5): 1 if it is present and 0 otherwise. Our integer program is

minimize $\sum_{\substack{R,C \text{ nonempty} \\ R,C\subseteq\mathcal{I}_n \\ \max R<\min C}} (|R|+|C|)x_{R,C}$
subject to the constraints
$x_{R,C} \in \{0,1\}$ for nonempty $R, C \subseteq \mathcal{I}_n$ and $\max R < \min C$
$\sum_{\substack{i\in R \\ j\in C}} x_{R,C} \geq 1$ for nonempty $R, C \subseteq \mathcal{I}_n$ and $\max R < \min C$.

The last constraint means that every string $a_i a_j$ with $i < j$ is covered by at least one concatenation of sets. Note that we write "≥ 1" in the last group of inequalities instead of "$= 1$", because we are not insisting that our regular expression be unambiguous.

For example, if $n = 3$ then the integer program is

minimize $2x_{0,1} + 2x_{0,2} + 2x_{1,2} + 3x_{01,2} + 3x_{0,12}$
subject to the constraints
$x_{0,1}, x_{0,2}, x_{1,2}, x_{01,2}, x_{01,2} \in \{0,1\}$
$x_{0,1} + x_{0,12} \geq 1$
$x_{0,2} + x_{01,2} + x_{0,12} \geq 1$
$x_{1,2} + x_{01,2} \geq 1.$

It is not difficult to see that our integer program, in fact, is the weighted set covering formulation, from Section 4, where the optimal value is $\mathsf{OR}_2(T_n)$ with T_n the $n \times n$ full triangular matrix, as in Section 5. So we can conclude from Corollary 5 that the smallest alphabetic length of a regular expression for the language L_n is $s(n) = n(\lfloor \log_2 n \rfloor + 2) - 2^{\lfloor \log_2 n \rfloor + 1}$.

In what follows, we illustrate the approach taken in the main text by formulating the linear relaxation of the integer program above and taking its dual. This follows Figure 2 in Section 4.

The integer program above is an instantiation of the one in Figure 2a. We now relax the constraints on the $x_{R,C}$ to be $0 \leq x_{R,C} \leq 1$. The dual linear program then has variables $y_{i,j}$ corresponding to the string $a_i a_j$, for $0 \leq i < j < n$; compare to Figure 2b. The corresponding dual, as in Figure 2c, is

maximize $\sum_{0\leq i<j<n} y_{i,j}$
subject to the constraints
$y_{i,j} \geq 0$ for $0 \leq i < j < n$
$\sum_{\substack{i\in R \\ j\in C}} y_{i,j} \leq |R| + |C|$ for nonempty $R, C \subseteq \mathcal{I}_n$ and $\max R < \min C$.

For example, for $n = 3$ the corresponding dual is

maximize $y_{0,1} + y_{0,2} + y_{1,2}$
subject to the constraints
$y_{0,1} \geq 0$
$y_{0,2} \geq 0$
$y_{1,2} \geq 0$
$y_{0,1} \leq 2$
$y_{0,2} \leq 2$
$y_{1,2} \leq 2$
$y_{0,1} + y_{0,2} \leq 3$
$y_{0,2} + y_{1,2} \leq 3.$

General connection

Whenever $L \subseteq \Sigma\Delta$ for the alphabets $\Sigma = \{1, \ldots, m\}$ and $\Delta = \{1, \ldots, n\}$, and M_L is its characteristic $m \times n$ matrix $M_{i,j} = 1$ iff $ij \in L$, then the following statements hold:

1. The value $\mathrm{OR}_2(M_L)$ coincides with the smallest possible alphabetic length of a regular expression for L.
2. The value $\mathrm{OR}_2(M_L)$ also coincides with the size of the smallest ε-free nondeterministic finite automaton (NFA) recognizing L.
3. The value $\mathrm{OR}(M_L)+m+n$ is an upper bound on the size of the smallest nondeterministic finite automaton with possible ε-transitions (ε-NFA) recognizing L.

The proof of the first statement follows the example above, and the last two statements can be found in [12].