Proof Complexity of Natural Formulas via Communication Arguments

Dmitry Itsykson
St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia

Artur Riazanov
St. Petersburg Department of Steklov Mathematical Institute of Russian Academy of Sciences, Russia

Abstract

A canonical communication problem Search (φ) is defined for every unsatisfiable CNF φ: an assignment to the variables of φ is partitioned among the communicating parties, they are to find a clause of φ falsified by this assignment. Lower bounds on the randomized k-party communication complexity of Search (φ) in the number-on-forehead (NOF) model imply tree-size lower bounds, rank lower bounds, and size-space tradeoffs for the formula φ in the semantic proof system $\mathrm{T}^{\mathrm{cc}}(k,c)$ that operates with proof lines that can be computed by k-party randomized communication protocol using at most c bits of communication [9]. All known lower bounds on Search (φ) (e.g. [1, 9, 13]) are realized on ad-hoc formulas φ (i.e. they were introduced specifically for these lower bounds). We introduce a new communication complexity approach that allows establishing proof complexity lower bounds for natural formulas.

First, we demonstrate our approach for two-party communication and apply it to the proof system $\mathrm{Res}(\oplus)$ that operates with disjunctions of linear equalities over $\mathbb{F}_2$ [14]. Let a formula PM_G encode that a graph G has a perfect matching. If G has an odd number of vertices, then PM_G has a tree-like $\mathrm{Res}(\oplus)$-refutation of a polynomial-size [14]. It was unknown whether this is the case for graphs with an even number of vertices. Using our approach we resolve this question and show a lower bound $2^{\Omega(n)}$ on size of tree-like $\mathrm{Res}(\oplus)$-refutations of $\mathrm{PM}_{K_{n+2,n}}$.

Then we apply our approach for k-party communication complexity in the NOF model and obtain a $\Omega\left(\frac{1}{k}2^{n/2k-3k/2}\right)$ lower bound on the randomized k-party communication complexity of Search $\left(\mathrm{BPHP}^{M}_{2^n}\right)$ w.r.t. to some natural partition of the variables, where $\mathrm{BPHP}^{M}_{2^n}$ is the bit pigeonhole principle and $M = 2^n + 2^{n(1-1/k)}$. In particular, our result implies that the bit pigeonhole requires exponential tree-like $\mathrm{Th}(k)$ proofs, where $\mathrm{Th}(k)$ is the semantic proof system operating with polynomial inequalities of degree at most k and $k = \mathcal{O}(\log^{1-\epsilon} n)$ for some $\epsilon > 0$. We also show that $\mathrm{BPHP}^{2^n+1}_{2^n}$ superpolynomially separates tree-like $\mathrm{Th}(\log^{1-\epsilon} m)$ from tree-like $\mathrm{Th}(\log m)$, where m is the number of variables in the refuted formula.

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

Keywords and phrases bit pigeonhole principle, disjointness, multiparty communication complexity, perfect matching, proof complexity, randomized communication complexity, Resolution over linear equations, tree-like proofs

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.3

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2020/184/`

Funding The research presented in Sections 3 and 4 is supported by Russian Science Foundation (project 18-71-10042).

Acknowledgements The authors are grateful to Anastasia Sofronova, Svyatoslav Gryaznov, Danil Sagunov, Petr Smirnov, Dmitry Sokolov, and Jakob Nordström for fruitful discussions and useful comments. We also thank the anonymous referees for useful comments and corrections. Dmitry Itsykson is a Young Russian Mathematics award winner and would like to thank sponsors and jury of the contest.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 3; pp. 3:1–3:34
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Propositional proof complexity studies proof systems that allow proving the unsatisfiability of Boolean CNF formulas. The main line of research in proof complexity is focused on refutation size lower bounds for different proof systems. This research activity is motivated by NP vs coNP question [3] as well as by studying properties of SAT-solvers. This paper develops the communication complexity approach to proof complexity lower bounds.

1.1 Communication complexity of search problems

In the classical communication settings, several participants collaborate to compute a function using a broadcast communication channel; each participant knows only a part of the input and the goal is to compute the function with the minimum number of transmitted bits. In the case of search problems, participants compute a relation $R \subseteq X \times Y$ instead of a function in the following sense: an input $x \in X$ is partitioned among the participants and they have to find $y \in Y$ such that $(x, y) \in R$. Analyzing the communication complexity of search problems is usually much harder than analyzing the communication complexity of functions. Unrestricted and monotone circuit depth of a Boolean function can be characterized in terms of the communication complexity of an appropriate search problem [17].

Every unsatisfiable CNF-formula φ defines a search problem $\mathrm{Search}(\varphi)$: the values of the variables of φ are partitioned between the parties of the protocol in some way, the participants are to find a clause of φ that is falsified by the values of the variables. This problem plays an important role in proof complexity.

One of the promising approaches for obtaining proof complexity lower bounds is the investigation of *dag-like* communication protocols [20, 33]. This approach allows proving lower bounds for proof systems operating with proof lines having small communication complexity in the appropriate communication model. Every refutation of a formula φ of size S can be translated to a dag-like communication protocol for $\mathrm{Search}(\varphi)$ of complexity $S \cdot C$, where C depends on the upper bound on the communication complexity of proof lines. Thus, lower bounds on the complexity of dag-like communication protocols imply lower bounds on the size of refutations. Nontrivial lower bounds on the size of dag-like protocols are currently known only for two-party deterministic and two-party real communication models. There are two known approaches for obtaining these lower bounds. The first is based on the correspondence between dag-like protocols and monotone Boolean/real circuits [20, 33, 11]. The second approach is lifting from the resolution width [7]. The mentioned lower bounds on dag-like communication imply lower bounds for Resolution [20], OBDD-based proof systems [21] (via deterministic protocols), and Cutting Planes [28, 10, 6, 7] (via real protocols).

Proving a superpolynomial lower bound for any of the models of dag-like communication protocols listed in the left column of Table 1 seems to be a very challenging open question. Such lower bounds would imply currently unknown superpolynomial lower bounds on the corresponding proof systems in the right column of the table.

In this paper, we deal with classical (tree-like) communication protocols. A lower bound on (tree-like) communication complexity of the problem $\mathrm{Search}(\varphi)$ in the model from the left column of Table 1 implies a lower bound on the size of tree-like refutations of φ in the corresponding proof system from the right column as well as a lower bound on the size of dag-like refutation of φ using small space (a size-space tradeoff [9, 12]). The usual strategy for obtaining lower bounds on the proof size via communication complexity is the following: by a tree-like refutation of φ of size S (or by a realization of a dag-like refutation of φ in size S within small space), one constructs a communication protocol for $\mathrm{Search}(\varphi)$ with

Table 1 Correspondence between communication models and proof systems.

Communication model	Proof systems
Randomized two-party protocols	Res($\oplus$) [15]. Proof lines in Res($\oplus$) are disjunctions of linear equations over $\mathbb{F}_2$.
Real k-party protocols in the number-on-forehead (NOF) model	Semantic Th($k-1$) [1]. Proof lines in Th($k-1$) are inequalities of the form $f(x_1, x_2, \ldots, x_n) \geq 0$, where f is a polynomial of degree at most $k-1$ with integer coefficients and Boolean variables.
Randomized k-party protocols in the NOF model	Semantic $\mathrm{T}^{\mathrm{cc}}(k, c)$. Proof lines in $\mathrm{T}^{\mathrm{cc}}(k, c)$ are arbitrary predicates that can be computed with k-party randomized communication cost at most c in the NOF model. $\mathrm{T}^{\mathrm{cc}}(k, c)$ for small c simulates Th($k-1$) and Res(PC_{k-1}). Proof lines in Res(PC_d) [22, 19] are disjunctions of polynomial equalities of the form $p(x_1, x_2, \ldots, x_n) = 0$, where p is a polynomial over $\mathbb{F}_2$ of degree at most d. Notice that Res(PC_1) coincides with Res($\oplus$).

communication complexity $\mathcal{O}(\log S \log\log S \cdot c)$[1] for an arbitrary partition of the variables of φ between the parties, where c is an upper bound for communication complexity of a proof line in the proof system in question. One then proceeds to prove a lower bound on the communication complexity of Search(φ) for some fixed partition of variables between the parties.

Proving lower bounds for the communication complexity of Search(φ) is not trivial since a lower bound on Search(φ) in the two-party deterministic communication model implies a lower bound on the monotone circuit depth for the corresponding monotone Boolean function [9, 29]. However, in the tree-like case good enough lower bounds are known for all models listed in the left column of Table 1. We discuss the strongest model, k-party randomized communication. Typically lower bounds on the communication complexity of Search(φ) are shown for artificial formulas φ that are constructed as follows: take a standard formula ψ and replace each of its variables with a function $g(y_1, y_2, \ldots, y_m)$ (also known as a gadget), where $y_1, y_2, \ldots, y_m$ are fresh variables; the result of this substitution is denoted by $\psi \circ g$. The variables of every gadget are partitioned among k parties. Beame, Pitassi and Segerlind [1] have shown a lower bound on the randomized k-party communication complexity of Search$(T(G) \circ \wedge_k)$, where $T(G)$ is an unsatisfiable Tseitin formula based on a special expander G and $\wedge_k$ is the conjunction of k variables, and the ith party has the ith argument of each instance of $\wedge_k$ written on their forehead.

Huynh and Nordström [12] have introduced a method to obtain a two-party randomized communication complexity lower bound for a search problem via lifting from search problems with large critical block sensitivity. Göös and Pitassi [9] have simplified and generalized this result to multiparty communication complexity and shown that if Search(φ) has large critical block sensitivity and a gadget g has a *versatile* property, then Search$(\varphi \circ g)$ has large randomized communication complexity. Although the construction of versatile functions is somewhat tricky, the proof of the lower bound is much simpler than the proofs from [1, 12].

[1] sometimes it can be improved to $\mathcal{O}(\log S \cdot c)$

There is an established stereotype that lower bounds on the randomized communication complexity of search problems are rather complicated and the resulting lower bounds for proof systems hold only for artificial formulas. In this paper, we break this stereotype and suggest an approach that allows obtaining lower bounds for natural families of formulas by reduction from randomized communication complexity. Moreover, our proofs are elementary.

In the first part of the paper, we demonstrate our method by proving an exponential lower bound on the size of tree-like Res($\oplus$)-refutations of the perfect matching principle, while the known lower bound techniques for tree-like Res($\oplus$) do not work for this formula. This lower bound is based on two-party communication complexity. In the second part of the paper, we apply our method to k-party communication complexity and prove a lower bound for communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{2^n+2^{n(1-1/k)}}\right)$, where $\mathrm{BPHP}_{2^n}^{M}$ denotes the bit pigeonhole principle stating that there are M distinct n-bit strings $s_1, \ldots, s_M$, every string s_i for $i \in [M]$ is partitioned into k almost equal sequential parts and the jth part of every string is written on the forehead of the jth party. In particular, the latter result implies that the bit pigeonhole principle is hard for tree-like $\mathrm{Th}(k)$, so it is the first natural hard instance.

1.2 Search problem $\oplus_k\mathrm{Search}\,(\varphi)$

To achieve our results we use the parity gadget, one of the simplest and the most natural gadgets. We then show how to get rid of this gadget using either properties of a proof system or properties of a family of formulas.

For an unsatisfiable CNF formula φ we define a k-party communication problem $\oplus_k\mathrm{Search}\,(\varphi)$ (usually denoted as $\mathrm{Search}\,(\varphi) \circ \oplus_k$) as follows: for every $i \in [k]$, the ith party has an assignment $\alpha_i \in \mathbb{F}_2^n$ written on the forehead, where n is the number of variables of φ. They are to find a clause of φ that is falsified by the assignment $\sum_{i=1}^{k} \alpha_i$.

It is easy to see that the communication complexity of $\mathrm{Search}\,(\varphi \circ \oplus_k)$ is at least the communication complexity of $\oplus_k\mathrm{Search}\,(\varphi)$, where $\oplus_k$ is the parity of the sum of k bits. However, the formula $\varphi \circ \oplus_k$ may have exponential size if φ contains a wide clause.

In Section 3 we observe the following lemma.

▶ **Lemma 1.** *If an unsatisfiable CNF-formula φ has a tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$ refutation of size S, then there exists a bounded-error randomized communication protocol for $\oplus_{d+1}\mathrm{Search}\,(\varphi)$ that transmits $\mathcal{O}(d \log S)$ bits.*

1.3 Perfect matching principle in tree-like $\mathrm{Res}(\oplus)$

One of the most important open questions in proof complexity is obtaining a superpolynomial lower bound for bounded-depth Frege with parity gates. Res($\oplus$) is a special case of this system and there are still no known superpolynomial lower bounds for its dag-like version. The first exponential lower bounds for tree-like Res($\oplus$) were proved by Itsykson and Sokolov [14, 15]. Itsykson and Sokolov have shown a lower bound $2^{\Omega(n)}$ on size of tree-like Res($\oplus$) refutations of Pigeonhole Principle (PHP_n^m) for arbitrary $m > n$ using generalized Prover-Delayer games. Oparin in [26] has shown a tight upper bound $2^{\mathcal{O}(n)}$ for such refutations. A lower bound $2^{\Omega(n)}$ for functional pigeonhole principle (FPHP_n^m) for $m = \mathcal{O}(n)$ can be shown using a connection between the size of tree-like Res($\oplus$) refutations and the degree of polynomial calculus refutations (over $\mathbb{F}_2$), observed by Garlik and Kolodziejczyk (see Section 7 of [8]; this method is described in details in [27]; an alternative explanation can be found in [19]). It is also worth mentioning the result of Krajicek (Theorem 18.6.4 from [23]) that formulas encoding Hall's theorem about matchings in bipartite graphs require exponential-size tree-like Res($\oplus$) refutations.

Let PM_G for a graph G encode the existence of a perfect matching in G. Itsykson and Sokolov [14, 15] have shown that for graphs with an odd number of vertices, PM_G has a polynomial-size tree-like $\mathrm{Res}(\oplus)$ refutation. The question about graphs with an even number of vertices remained open; we resolve it in this paper.

Let $K_{m,n}$ be the complete bipartite graphs with parts of size m and n respectively. In Section 4 we prove the following theorem.

▶ **Theorem 2.** *The size of a tree-like* $\mathrm{Res}(\oplus)$ *refutation of* $\mathrm{PM}_{K_{n+2,n}}$ *is* $2^{\Omega(n)}$.

Notice that since PHP^m_n is a weakening of $\mathrm{PM}_{K_{m,n}}$, Oparin's upper bound for PHP^m_n [26] implies that the obtained lower bound is tight up to a constant in the exponent.

The formula $\mathrm{PM}_{K_{n+2,n}}$ (however, in a different encoding) has a constant-degree derivation in Nullstellensatz over $\mathbb{F}_2$ [2]. $\mathrm{PM}_{K_{n+2,n}}$ may be refuted as follows: compute the number of edges in the matching modulo 4 in two different ways, on the one hand it is $n \bmod 4$ and on the other hand it is $(n+2) \bmod 4$. This yields a low-degree Nullstellensatz refutation since the function MOD_4 has a representation as a polynomial of degree 3, see Lemma 8.7 of [2] for details. Thus, Theorem 2 can not be proved via the same reduction to the Polynomial Calculus degree as it can be done for FPHP^m_n.

Since $\mathrm{PM}_{K_{n+2,n}}$ has a tree-like Cutting Planes refutation of polynomial size and with polynomial coefficients, the problem $\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$ has communication complexity $\mathcal{O}(\log n)$ for any partition and thus can not yield a superpolynomial lower bound on size of tree-like $\mathrm{Res}(\oplus)$ refutations. *Therefore the methods previously used to establish tree-like* $\mathrm{Res}(\oplus)$ *lower bounds fail for* $\mathrm{PM}_{K_{n+2,n}}$.

To establish this lower bound we employ an idea similar to the one used in [30] to show monotone circuit depth lower bound for matching.

Proof sketch of Theorem 2. By Lemma 1 it is sufficient to show a lower bound $\Omega(n)$ on the two-party bounded-error randomized communication complexity of $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$. We show this lower bound via probabilistic reduction from the set disjointness problem. Recall that in the set disjointness problem DISJ_n Alice and Bob have strings $x, y \in \{0,1\}^n$ respectively and they want to verify that there are no $i \in [n]$ such that $x_i = y_i = 1$. It is known that two-party bounded-error randomized communication complexity of DISJ_n is $\Omega(n)$ [16]. Let $G_0(V, E_1)$ and $G_1(V, E_1)$ be graphs on the same set of vertices V; we define $G_0 \oplus G_1$ as a graph on V with edges $E_1 \oplus E_2$, where $\oplus$ denotes the symmetric difference.

We now describe the reduction from DISJ_n to $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$. Before starting the communication, each of the parties constructs two graphs: Alice constructs $A(0)$ and $A(1)$, Bob constructs $B(0)$ and $B(1)$ that are shown in Figure 1. These four graphs are bipartite graphs on 8 vertices, 4 vertices in each part and the parts coincide for all the graphs. These graphs have the following property: for $a, b \in \{0,1\}$ the graph $A(a) \oplus B(b)$ is a perfect matching iff at least one of a and b is zero. The graph $A(1) \oplus B(1)$ has two connected components, the first component consists of a single vertex from the first part connected with three vertices from the second part, the second connected component consists of a single vertex from the second part connected with three vertices from the first part.

For each $i \in [n]$ Alice and Bob create new 8 vertices; Alice builds the graph $A(x_i)$ on these vertices and Bob builds the graph $B(y_i)$ on these vertices. Thus, Alice and Bob construct two bipartite graphs G_A and G_B with $4n$ vertices in each part such that $G_A \oplus G_B$ is a perfect matching iff $\mathrm{DISJ}_n(x, y) = 1$. Additionally, Alice and Bob add three vertices to the first part and one vertex to the second part of $G_A \oplus G_B$ connecting the latter with the three vertices added to the first part. Let us denote the resulting graph by H. Let $H = H_A \oplus H_B$, where H_A is known to Alice and H_B is known to Bob. An example of the

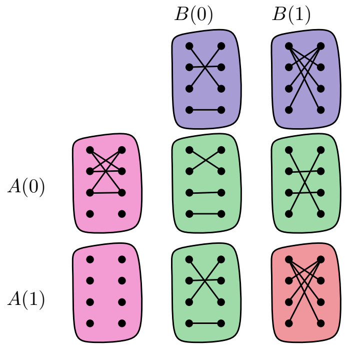

Figure 1 The graphs $A(0), A(1)$, $B(0)$, and $B(1)$ and their pairwise symmetric differences. Only $A(1) \oplus B(1)$ is not a matching.

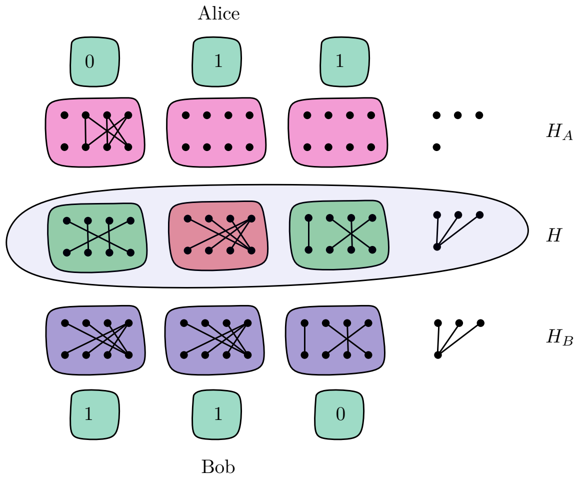

Figure 2 The construction of the graphs H_A, H_B and H for $x = (0, 1, 1)$; $y = (1, 1, 0)$.

resulting graphs is shown in Figure 2. Alice and Bob shuffle the vertices in each part of their graphs according to a permutation generated using public random bits and get graphs H'_A and H'_B. As a result, in the shuffled graph $H' = H'_A \oplus H'_B$ the violation of the perfect matching principle artificially added by Alice and Bob is indistinguishable from a violation that appears because of $\mathrm{DISJ}_n(x, y) = 0$. After that Alice and Bob run the communication protocol for $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{4n+3,4n+1}}\right)$. If the protocol returns a clause corresponding to the artificially added contradiction, Alice and Bob return 1; otherwise, they return 0. By repeating the whole protocol multiple times one can reduce the error probability. ◀

1.4 Bit pigeonhole principle

1.4.1 Bit pigeonhole principle with $\oplus$-gadget

In Section 5 we apply our lower bound technique for k-party communication in the number-on-forehead model. We consider the bit pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ that encodes in CNF that there are m pairwise distinct strings from $\{0, 1\}^\ell$. This formula is unsatisfiable for $m > 2^\ell$.

▶ **Theorem 3.** *Let ℓ and k be natural numbers such that $2 \leq k \leq \ell - 7$. Then the randomized communication complexity of $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell+2^k}_{2^\ell}\right)$ in the k-party NOF model is $\Omega\left(\frac{2^{\ell/2}}{k 2^{3k/2}}\right)$. For $k = 2$ the stronger bound $\Omega\left(2^\ell\right)$ holds.*

Proof idea. The proof follows the same plan as the communication complexity lower bound in Theorem 2. In Subsection 5.1 we consider a decision problem $\mathrm{Distinct}_{k,\ell}$ that is similar to the search problem $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell}_{2^\ell}\right)$. Let each of k parties have a $2^\ell \times \ell$ matrix over $\mathbb{F}_2$ on the forehead. The goal is to determine whether the rows of the sum of these matrices are distinct. Recall that the unique disjointness $\mathrm{UDISJ}_{k,n}$ is the promise version of the k-party set disjointness: the ith of k parties has a string $x^{(i)}$ from $\{0, 1\}^n$ on the forehead, they are to verify that there is no $j \in [n]$ such that $x^{(i)}_j = 1$ for all $i \in [k]$ under the promise that there is at most one such index j. We describe a randomized reduction from $\mathrm{UDISJ}_{k,2^{\ell-k}+1}$ to $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell}_{2^\ell}\right)$ and then use the known lower bound on the communication complexity of the former problem [32]. First, we reduce $\mathrm{UDISJ}_{k,2^{\ell-k}}$ to the problem $\mathrm{Distinct}_{k,\ell}$: the ith of the parties of the UDISJ protocol generates a matrix D_i of size $2^\ell \times \ell$ such that the matrix $\sum_{i=1}^k D_i$ contains a pair of equal rows iff $\mathrm{UDISJ}_{k,2^{\ell-k}}$ evaluates to 0. Moreover, the matrix $\sum_{i=1}^k D_i$ has additional properties:

- each of the $2^{\ell-k}$ bits of UDISJ correspond to a block of 2^k rows of the matrix $\sum_{i=1}^k D_i$ such that any two rows from different blocks are distinct;
- if the common 1-bit of the inputs of UDISJ has the index $j \in [2^{\ell-k}]$, then the block corresponding to the bit j contains each of its rows exactly twice (all the other blocks have distinct rows).

In Subsections 5.2 and 5.3 we adapt this reduction for $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)$. We add an additional (fake) block to each of the matrices D_i such that the matrix $\sum_{i=1}^k D_i$ has the following property: every row of this new block appears in it exactly twice and does not appear anywhere else. Using randomization we make sure that the new artificially added row collisions from the fake block are indistinguishable from the collisions coming from the initial (genuine) blocks corresponding to the bits of UDISJ. Finally, if UDISJ evaluates to 1 then all the collisions are artificially added; if UDISJ evaluates to 0, then with a significant probability the protocol solving $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)$ finds a pair of equal rows coming from a genuine block. ◀

Theorem 3 and Lemma 1 immediately imply the lower bound $\exp\left(\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)\right)$ on the size of tree-like $\mathrm{Res}\,(\mathrm{PC}_{k-1})$ refutations of $\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}$ (for $k = 2$ the stronger lower bound $\Omega(2^\ell)$ holds).

1.4.2 Bit pigeonhole without $\oplus$-gadget

In Section 6 we present a pretty simple and nice reduction from $\oplus_k\mathrm{Search}\,(\mathrm{BPHP}_{2^n}^m)$ to $\mathrm{Search}\left(\mathrm{BPHP}_{2^{kn}}^{m\cdot 2^{(k-1)n}}\right)$. Here we describe this reduction for $k = 2$. For a larger k the proof is essentially the same. Let us reduce $\oplus_2\mathrm{Search}\,(\mathrm{BPHP}_{2^n}^m)$ to $\mathrm{Search}\left(\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}\right)$. We denote the input of Alice in $\oplus_2\mathrm{Search}\,(\mathrm{BPHP}_{2^n}^m)$ as $a_1, \dots, a_m \in \mathbb{F}_2^n$ and the input of Bob as $b_1, \dots, b_m \in \mathbb{F}_2^n$. Their goal is to find a clause of $\mathrm{BPHP}_{2^n}^m$ falsified by the assignment $a_1 + b_1, \dots, a_m + b_m$. Observe that given $i \neq j \in [m]$ such that $a_i + b_i = a_j + b_j$ they can find a falsified clause transmitting additional $\mathcal{O}(n)$ bits. For each $i \in [m]$, Alice and Bob generate 2^n strings from $\mathbb{F}_2^n$: Alice generates $a_i + z$ for each $z \in \mathbb{F}_2^n$ and Bob generates $b_i + z$ for each $z \in \mathbb{F}_2^n$. For each pair of strings $a_i + z$ and $b_i + z$ their sum coincides with $a_i + b_i$. Alice and Bob run the protocol for $\mathrm{Search}\left(\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}\right)$ on an input where each line has the form $(a_i + z, b_i + z)$ for each $i \in [m]$ and $z \in \mathbb{F}_2^n$. Given a falsified clause of $\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}$ on this input they determine the lines $(a_i + z, b_i + z)$ and $(a_j + z', b_j + z')$ that are equal to each other. Then $a_i + b_i = a_j + b_j$ and $i \neq j$ since each pair $(i, z) \in [m] \times \mathbb{F}_2^n$ is used by Alice and Bob exactly once.

Together with Theorem 3 this yields the following theorem.

▶ **Theorem 4.** *For $n \geq k(k+7)$ the randomized k-party communication complexity of* $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{2^n+2^{n+k-\lfloor n/k\rfloor}}\right)$ *is* $\Omega\left(\frac{1}{k}2^{n/2k-3k/2}\right)$, *where every string of* BPHP *is partitioned into k almost equal contiguous parts such that jth party has the jth part of every string on its forehead. For $k = 2$ the bound can be improved up to* $\Omega\left(2^{n/2}\right)$.

In particular, Theorem 4 implies the lower bound $\exp\left(2^{\Omega(n/k)}\right)$ on the size of tree-like $\mathrm{T^{cc}}(k, c)$ (and $\mathrm{Th}(k-1)$) refutations of $\mathrm{BPHP}_{2^n}^{2^n+2^{n+k-\lfloor n/k\rfloor}}$.

Hrubes and Pudlák [10] proved a lower bound on the complexity of dag-like two-party real communication protocols for Search $(\mathrm{BPHP}^m_{2^\ell})$ with the same variable partition, where $m > 2^\ell$ is arbitrary. Formally their and our results are incomparable. On the one hand, the result of Hrubes and Pudlák holds for dag-like protocols and arbitrary weak bit pigeonhole principle, on the other hand, we use a stronger (randomized) model and the statement holds for the multiparty communication as well.

In addition, we show an upper bound on the communication complexity of Search $(\mathrm{BPHP}^m_{2^\ell})$. The gap between the upper and the lower bound for $k > 2$ is quadratic. For $k = 2$ the bounds coincide up to a logarithmic factor.

▶ **Proposition 5.** *For $M > 2^n$ and $k \in \{2, 3, \dots, n\}$ there exists a* deterministic *NOF communication protocol for* Search $\left(\mathrm{BPHP}^M_{2^n}\right)$ *with variables partitioned as in Theorem 4 transmitting $\mathcal{O}\left(2^{\lceil n/k\rceil} \cdot \log M\right)$ bits.*

Our lower bound on the k-party communication complexity of Search (BPHP^m_n) is non-trivial for $k \le \log^{1-\varepsilon} n$ for $\varepsilon > 0$. This lower bound implies a superpolynomial lower bound on the size of tree-like $\mathrm{Th}(k)$-refutations of BPHP^m_n for such k. We show that there exists a short tree-like $\mathrm{Th}(\log n)$ refutation:

▶ **Proposition 6.** *For $m > 2^\ell$ there exists a tree-like* $\mathrm{Th}(\ell)$ *refutation of* $\mathrm{BPHP}^m_{2^\ell}$ *of size $\mathcal{O}(m^2 \cdot 2^\ell)$.*

Proposition 6 and the result of Hrubes and Pudlák [10] imply that tree-like $\mathrm{Th}(\log n)$, where n is the number of variables of the refuted formula can not be simulated by *dag-like* $\mathrm{Th}(1)$. Theorem 4 and Proposition 6 imply that the bit pigeonhole principle $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ separates [2] tree-like $\mathrm{Th}(\log n)$ from tree-like $\mathrm{Th}(k)$ for $k \le \log^{1-\varepsilon} n$.

1.5 Open questions

1. Is it possible to prove lower bounds on the randomized communication complexity of $\oplus_2$Search (PM_G) for *constant-degree* graphs G? An $\Omega(n)$ lower bound would improve the best known $\Omega(n/\log n)$ lower bound on the two-party communication complexity of a Search (φ) problem, where n is the number of variables.
2. Is it true that our results extend to Res (PC_d) over arbitrary finite fields?
3. Is our lower bound for tree-like $\mathrm{Th}(k)$ refutation of $\mathrm{BPHP}^m_{2^n}$ tight? Such upper bound would imply a superpolynomial separation between *tree-like* $\mathrm{Th}(k)$ and *dag-like* cutting planes due to the lower bound by [10] as well as separations between tree-like $\mathrm{Th}(k)$ for different values of k.
4. Can we show a lower bound on the communication complexity of the search problem for weaker versions of $\mathrm{BPHP}^M_{2^n}$, for example with $M = 2^{n+1}$?

[2] The formula $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ uses $n = (2^\ell + 1)\ell$ variables. By Proposition 6, there is a tree-like $\mathrm{Th}(\log n)$ refutation of size $\mathrm{poly}(n)$. By Theorem 4, the size of any tree-like $\mathrm{Th}(\log^{1-\epsilon} n)$ refutation is at least $\exp(\exp(\Omega(\log^\epsilon n)))$; the latter grows superpolynomially in n.

2 Preliminaries

Notations

We use the following notation: $[n] = \{1, 2, \dots, n\}$. Let $S^{n\times m}$ denote the set of matrices of size $n \times m$ with elements from S. We denote by $\mathbf{0}_{n\times m}$ the zero matrix of size $n \times m$ and by $\mathbf{1}_{n\times m}$ the matrix of the same size containing only ones. For square matrices $A_1, \dots, A_k$ we denote a diagonal block matrix with blocks $A_1, \dots, A_k$ by $\mathrm{diag}(A_1, \dots, A_k)$. For $x \in \{0, 1, \dots, 2^k - 1\}$ we denote a vector $(a_0, \dots, a_{k-1}) \in \{0, 1\}^k$ such that $x = \sum_{i=0}^{k-1} a_i 2^i$ by $\mathtt{bin}_k(x)$, i.e. $(a_0, \dots, a_{k-1})$ is the *reversed* binary representation of x. For vectors $v_1, \dots, v_n$ from a vector space over a field $\mathbb{F}$ we denote their linear span by $\mathrm{Span}(v_1, \dots, v_n)$.We use coordinate-wise comparison of strings from $\{0, 1\}^n$, i.e. for $x, y \in \{0, 1\}^n$ we write $x \le y$ iff $x_i \le y_i$ for each $i \in [n]$. We denote the set of variables of a CNF-formula φ by $\mathrm{Vars}(\varphi)$.

Communication complexity

We briefly recall some notions of communication complexity. For formal definition and details we refer to [24].

In the classic two-party randomized communication protocol with public randomness, Alice and Bob cooperate to compute a relation $Q \subseteq X \times Y \times Z$: Alice has an input $x \in X$ and Bob has an input $y \in Y$, their goal is to compute $z \in Z$ such that $(x, y, z) \in Q$. We assume that Alice and Bob have access to an arbitrary large random string of bits that is common for Alice and Bob. Let for every $x \in X$ and $y \in Y$, $R^{\delta}_{pub}(Q, x, y)$ denote the minimal number of bits Alice and Bob need to transmit between each other so they both find a $z \in Z$ such that $(x, y, z) \in Q$ with probability at least $1 - \delta$ taken over the values of the common random string. And $R^{\delta}_{pub}(Q) := \max_{x\in X, y\in Y} R^{\delta}_{pub}(Q, x, y)$.

We also consider multiparty communication protocols in the number on forehead (NOF) model that extends two-party protocols for an arbitrary number of parties. In this setting k parties cooperate to compute a relation $Q \subseteq X_1 \times X_2 \times \dots \times X_k \times Y$. The ith party has $x_i \in X_i$ written on their forehead so they know all x_j for $j \ne i$, their goal is to compute $y \in Y$ such that $(x_1, x_2, \dots, x_k, y) \in Q$. The parties communicate by taking turns broadcasting messages to all other parties until all parties learn the value of $y \in Y$ such that $(x_1, \dots, x_k, y) \in Q$. In this model we also assume that all parties have access to a common random string of bits. Let $R^{\delta}_{pub}(Q, x_1, \dots, x_k)$ for $x_1 \in X_1, \dots, x_k \in X_k$ denote the minimal total number of bits transmitted until each party learns $y \in Y$ such that $(x_1, \dots, x_k, y) \in Q$ with probability at least $1 - \delta$ taken over the set of values of the random string of bits. Also, let $R^{\delta}_{pub}(Q) := \max_{x_1\in X_1,\dots,x_k\in X_k} R^{\delta}_{pub}(Q, x_1, \dots, x_k)$.

Let f be a function from $X_1 \times X_2 \times \dots \times X_k \to Y$. Then $R^{\delta}_{pub}(f)$ denotes $R^{\delta}_{pub}(Q_f)$, where $Q_f = \{(x_1, x_2, \dots, x_k, y) \mid f(x_1, \dots, x_k) = y\}$.

We prove communication complexity lower bounds by reduction from different versions of the set disjointness problem. $\mathrm{DISJ}_{k,n}$ is a function $\{0, 1\}^{kn} \to \{0, 1\}$ such that for every $x_1, \dots, x_k \in \{0, 1\}^n$ the following holds: $\mathrm{DISJ}_{k,n}(x_1, \dots, x_k) = \bigwedge_{j=1}^{n} \underbrace{\neg\left(\bigwedge_{i=1}^{k} (x_i)_j\right)}_{NAND}$.

Let us define the communication promise problem $\mathrm{UDISJ}_{k,n}$ in the k-party NOF model. For each $i \in [k]$ the string x_i is written on the forehead of the ith party, it is guaranteed that there exists at most one index $j \in [n]$ such that for every $i \in [k]$, $(x_i)_j = 1$. The goal is to compute $\mathrm{DISJ}_{k,n}(x_1, \dots, x_k)$.

▶ **Theorem 7** ([31, 32]). $R^{1/3}_{pub}(\mathrm{UDISJ}_{k,n}) = \Omega\left(\frac{\sqrt{n}}{2^k k}\right)$.

For $k = 2$ we omit the first index: $\mathrm{DISJ}_n = \mathrm{DISJ}_{2,n}$; in this case Theorem 7 may be improved.

▶ **Theorem 8** ([16]). $R^{1/3}_{pub}(\mathrm{DISJ}_n) \geq R^{1/3}_{pub}(\mathrm{UDISJ}_{2,n}) = \Omega(n)$.

Proof complexity

We consider refutational proof systems for the language of unsatisfiable CNF-formulas UNSAT. A refutation of $\varphi \in$ UNSAT in a proof system Π is a sequence of Boolean functions (proof lines) such that each proof line either represents a clause of φ or derived from previous proof lines in the sequence via some sound inference rules. The last line of the proof is identically zero function. A proof system Π is defined by a representation of proof lines and by a set of admissible inference rules. It is required that the inference rules are polynomially verifiable i.e. there exists an algorithm that checks whether it is legitimate to derive a line L_0 from the lines $L_1, \dots, L_k$.

For example, in the Resolution proof lines are represented by clauses and the only inference rule is the resolution rule that allows deriving a clause $A \vee B$ from the clauses $A \vee x$ and $A \vee \neg x$.

The *size* of a proof is the total size of all representations of proof lines in the proof. The *length* of a proof is the number of proof lines in it.

A tree-like proof is such a proof that every its line can be used as a premise of a rule at most once. For each proof system, we can also consider its tree-like version where all proofs are constrained to be tree-like.

We also consider *semantic* refutational proof systems, where we drop the requirement for polynomial verification of inference rules i.e. we allow to derive any sound consequence from the premises. For such systems it is crucial to bound fan-in i.e. the number of the premises from which each proof line can be derived, otherwise, it would be possible to derive a contradiction from the clauses of the initial formula immediately. For example, it is well-known that Resolution is polynomially equivalent to a semantic proof system with fan-in 2 operating with clauses.

A lower bound on the proof size in a semantic proof system implies a lower bound on the proof size in its syntactic counterpart because a syntactic proof is always a semantic proof that operates with the same class of proof lines.

We define semantic $\mathrm{Res}(\oplus)$ as a semantic proof system with fan-in 2 that operates with linear clauses. A linear clause is a disjunction of linear equations over $\mathbb{F}_2$: $\bigvee_{i=1}^{k}(f_i = a_i)$, where f_i is a linear form over $\mathbb{F}_2$ and $a_i \in \mathbb{F}_2$. Notice that an ordinary clause $\bigvee_{i\in P} x_i \vee \bigvee_{j\in N} \neg x_j$ can be represented by the linear clause $\bigvee_{i\in P}(x_i = 1) \vee \bigvee_{j\in N}(x_j = 0)$. For definition of syntactic version of $\mathrm{Res}(\oplus)$ we refer to [15]; it is also proved there that syntactic and semantic $\mathrm{Res}(\oplus)$ are polynomially equivalent.

We define semantic $\mathrm{Res}\,(\mathrm{PC}_d)$ as a semantic proof system with fan-in 2 that operates with disjunctions of equations of type $f = 0$, where f is a degree-d polynomial over $\mathbb{F}_2$. Notice that semantic $\mathrm{Res}\,(\mathrm{PC}_1)$ is exactly semantic $\mathrm{Res}(\oplus)$. For the definition of the syntactic version of $\mathrm{Res}\,(\mathrm{PC}_d)$ we refer to [19].

Following [1] we define $\mathrm{Th}(k)$ as a semantic proof system with fan-in 2 that operates with polynomial inequalities $g \geq 0$, where g is a polynomial of degree at most k with integer coefficients and Boolean variables. A clause $\bigvee_{i\in P} x_i \vee \bigvee_{j\in N} \neg x_j$ can be represented by an inequality $\sum_{i\in P} x_i + \sum_{j\in N}(1 - x_j) - 1 \geq 0$.

Proof complexity and communication complexity

For an unsatisfiable CNF-formula φ we define the communication problem $\mathrm{Search}(\varphi)$. $\mathrm{Search}(\varphi)$ is the following problem: given an assignment of the variables of the unsatisfiable CNF φ, find a clause that is falsified by this assignment. It is assumed that variables of φ are somehow partitioned between the parties.

Following the paper [9] we consider a semantic proof system $\mathrm{T}^{\mathrm{cc}}(k, c)$ that models many interesting syntactic and semantic proof systems. The proof lines in $\mathrm{T}^{\mathrm{cc}}(k, c)$ can be arbitrary Boolean functions having the following property: for every proof line C and every partition of variables of C between k parties, the NOF k-party randomized communication complexity of C is at most c w.r.t. this partition. We also define a semantic proof system $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ that is a subsystem of $\mathrm{T}^{\mathrm{cc}}(k, c)$ with the restriction that a communication protocol for proof lines must have a one-sided error: if the value of a proof line is zero, then the protocol should return zero with probability 1.

For example, $\mathrm{T}^{\mathrm{cc}}(2, 2)$ simulates Resolution; $\mathrm{T}^{\mathrm{cc}}(2, \mathcal{O}(1))$ simulates $\mathrm{Res}(\oplus)$ [22, 15]; $\mathrm{T}^{\mathrm{cc}}(k, \mathcal{O}(k^3 \log^2 n))$, where n is the number of variables in a refuted formula, simulates $\mathrm{Th}(k-1)$ [9]. In Section 3 we show that $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(d+1, \mathcal{O}(1))$ simulates $\mathrm{Res}(\mathrm{PC}_d)$.

The following connection between the communication complexity of $\mathrm{Search}(\varphi)$ and *tree-like* proof complexity of φ is known.

▶ **Lemma 9** ([1, 9]). *If a CNF formula φ has a tree-like $\mathrm{T}^{\mathrm{cc}}(k, c)$ refutation of length ℓ then, over any k-partition of the variables, there is a randomized bounded-error k-party NOF protocol for $\mathrm{Search}(\varphi)$ with communication cost $\mathcal{O}(c \cdot \log \ell \log \log \ell)$.*

In Section 3 we show that for $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ the bound can be improved, see Remark 14.

Basic formulas

A CNF formula PHP^m_n encodes the pigeonhole principle; PHP^m_n states that it is possible to put m pigeons into n holes such that every pigeon flies to at least one hole and at most one pigeon flies to each hole. PHP^m_n depends on variables $p_{i,j}$ for $i \in [m]$ and $j \in [n]$ and $p_{i,j} = 1$ iff the i-th pigeon flies to the j-th hole. PHP^m_n is the conjunction of $\frac{m(m-1)n}{2}$ hole axioms and m pigeons axioms. For every $i \in [m]$ PHP^m_n contains a pigeon axiom $(p_{i,1} \vee p_{i,1} \vee \cdots \vee p_{i,n})$. And for every $j \in [n]$ and every $k \neq \ell \in [n]$, PHP^m_n contains a hole axiom $(\neg p_{k,j} \vee \neg p_{\ell,j})$. PHP^m_n is unsatisfiable iff $m > n$.

For an undirected graph $G(V, E)$, the formula PM_G encodes in CNF that G has a perfect matching. The formula PM_G has $|E|$ variables, each of them corresponds to an edge of G, x_e is the variable corresponding to $e \in E$.

$$\mathrm{PM}_G = \bigwedge_{v \in V} \left(\left(\bigvee_{e \text{ is incident to } v} x_e \right) \wedge \bigwedge_{e_1 \neq e_2 \text{ are incident to } v} (\neg x_{e_1} \vee \neg x_{e_2}) \right).$$

PM_G is unsatisfiable iff G does not have a perfect matching.

▶ **Theorem 10** ([26]). *Let G be a graph with n vertices, which has no perfect matching. Then the formula PM_G has a tree-like $\mathrm{Res}(\oplus)$ refutation of size $2^{\mathcal{O}(n)}$.*

▶ **Proposition 11** ([14]). *Let G be a graph with an odd number of vertices. Then the formula PM_G has a tree-like $\mathrm{Res}(\oplus)$ refutation of size $\mathrm{poly}(n)$.*

The binary pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ states that there are m different ℓ-bit binary strings $s_1, s_2, \ldots, s_m$. $\mathrm{BPHP}^m_{2^\ell}$ has $m\ell$ variables corresponding to the bits of s_i for $i \in [m]$. Then $\mathrm{BPHP}^m_{2^\ell} = \bigwedge_{i \neq j \in [m]} s_i \neq s_j$, where the predicate $s_i \neq s_j$ is encoded as a 2ℓ-CNF formula of size 2^ℓ as follows: $\bigwedge_{\alpha\in\{0,1\}^\ell}(s_i \neq \alpha \vee s_j \neq \alpha)$; notice that the predicate $(s_i \neq \alpha \vee s_j \neq \alpha)$ can be represented by a clause with 2ℓ literals. If $m > 2^\ell$, then the formula $\mathrm{BPHP}^m_{2^\ell}$ is unsatisfiable.

Let φ be a CNF formula with n variables, and $g : \{0,1\}^k \to \{0,1\}$ be a Boolean function. Then $\varphi \circ g$ denotes a CNF formula on kn variables that represents $\varphi(g(\overrightarrow{x_1}), g(\overrightarrow{x_2}), \dots, g(\overrightarrow{x_n}))$, where $\overrightarrow{x_i}$ denotes a vector of k new variables. $\varphi \circ g$ is constructed by applying the substitution to every clause C of φ and converting the resulting function $C \circ g$ to CNF in some fixed way.

3 Communication protocols from tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$ proofs

Let φ be an unsatisfiable CNF formula with n variables. Let us define the communication problem $\oplus_k\mathrm{Search}\,(\varphi)$ with k parties as follows. Assume that the ith party has an assignment $\alpha_i \in \{0,1\}^n$ written on the forehead. They aim to find a clause of φ falsified by the assignment $\sum_{i=1}^k \alpha_i$ (all sums of boolean vectors are computed modulo 2).

▶ **Lemma 1.** *Let φ be an unsatisfiable CNF formula. If there exists a tree-like* $\mathrm{Res}\,(\mathrm{PC}_d)$ *proof of φ of length m, then $R^{1/3}_{pub}(\oplus_{d+1}\mathrm{Search}\,(\varphi)) = \mathcal{O}(d \cdot \log m)$.*

A slightly weaker version of the following lemma was implicitly proved in [15]:

▶ **Lemma 12** (see proof of Theorem 3.11 from [15])**.** *Let T be a binary tree with m vertices such that the ith vertex is labeled with $a_i \in \{0,1\}$ with the* hereditary property*: for each inner vertex i with direct descendants c_1 and c_2, if $a_i = 1$, then $a_{c_1} = 1$ or $a_{c_2} = 1$. We also assume that if r is the root of T, then $a_r = 1$. Assume that we have a one-sided bounded error oracle access to a_i i.e. if we request a value of a_i and $a_i = 0$ we get 1 with probability at most $\frac{1}{2}$ and 0 with probability at least $\frac{1}{2}$; if $a_i = 1$ we get 1 with probability 1. Then there exists an algorithm $\mathcal{A}$ that with probability at least $\frac{2}{3}$ returns a leaf ℓ of T with $a_\ell = 1$ and makes $\mathcal{O}(\log m)$ oracle queries to $a_1, \dots, a_m$.*

Proof. See Appendix A. ◀

Proof of Lemma 1. Let $F_1, \dots, F_m$ be a tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$-refutation of φ with the underlying tree T, where vertices of T are identified with $[m]$. Then the leaves of T correspond to the clauses of φ and m is the root of T.

Let $\alpha_1, \dots, \alpha_{d+1}$ be the assignments written on the foreheads of $d+1$ parties. Let $\alpha = \sum_{i=1}^{d+1} \alpha_i$. Let $a_i = 1$ iff α falsifies F_i for $i \in [m]$. Then $a_m = 1$ since F_m is identically false. For any inner node v of T, if $a_v = 1$ then for the direct descendants of v, c_1 and c_2 either $a_{c_1} = 1$ or $a_{c_2} = 1$. In the next paragraphs we show that for any $i \in [m]$ there exists a NOF $(d+1)$-party protocol that computes a_i given that for each $j \in [d+1]$ the jth party has α_j written on their forehead such that

- the protocol transmits $\mathcal{O}(d)$ bits;
- the protocol has one-sided bounded error: if $a_i = 1$ then the protocol returns 1 with probability 1 and if $a_i = 0$ the protocol returns 0 with probability at least $\frac{1}{2}$.

Then we use this protocol to compute a_i as an oracle in the algorithm given by Lemma 12 and thus show that there is a NOF $(d+1)$-party protocol computing $\oplus_{d+1}\mathrm{Search}\,(\varphi)$ with communication cost $\mathcal{O}(d \log m)$.

Now we show that for every $\ell \in [m]$, $F_\ell(\alpha)$ can be computed by a $(d+1)$-party NOF protocol with one-sided error using $\mathcal{O}(d)$ bits of communication. Let $F_\ell = \bigvee_{j=1}^t (f_j = 1)$, where $f_1, \dots, f_t$ are polynomials over $\mathbb{F}_2$ of degree at most d. Let $z_1, \dots, z_n$ be the variables of φ. Let us introduce new variables $y_{1,1}, \dots, y_{1,n}, \dots, y_{d+1,1}, \dots, y_{d+1,n}$ and assume that for each $i \in [d+1]$ the ith party has the value of variables $y_{i,1}, y_{i,2}, \ldots, y_{i,n}$ written on the forehead or in other words α_i assigns values of $y_{i,1}, y_{i,2}, \ldots, y_{i,n}$. Let $\bar{f}_j$ denote f_j after substitution $z_\ell := y_{1,\ell} + y_{2,\ell} + \ldots + y_{d+1,\ell}$ for $\ell \in [n]$; $j \in [t]$. Since for all $j \in [t]$, $\deg \bar{f}_j = \deg f_j \le d$, we can represent $\bar{f}_j = \bar{f}_j^{(1)} + \ldots + \bar{f}_j^{(d+1)}$ such that $\bar{f}_j^{(s)}$ does not contain variables $y_{s,1}, \ldots, y_{s,n}$ for each $s \in [d+1]$. Then the ith party can compute $\bar{f}_1^{(i)}(\alpha_1, \ldots, \alpha_{d+1}), \ldots, \bar{f}_t^{(i)}(\alpha_1, \ldots, \alpha_{d+1})$. Notice that $F_\ell = \neg\left(\bigwedge_{j=1}^t (f_j = 0)\right)$.

The final step of the protocol exploits the idea used to construct a short randomized communication protocol for equality. Take a random uniformly distributed vector $(e_1, \ldots, e_t) \in \mathbb{F}_2^t$. Then all parties compute $\sum_{j=1}^t e_j f_j(\alpha) = \sum_{i=1}^{d+1} \underbrace{\sum_{j=1}^{t} e_j \bar{f}_j^{(i)}}_{i\text{th party}}$ with $\mathcal{O}(d)$ bits of communication and the protocol halts.

To bound the error probability we use the following well-known statement:

▶ **Proposition 13** (Random subsum principle)**.** *For any $x \in \mathbb{F}_2^k \setminus \{0^k\}$,*

$$\Pr_{y \leftarrow \mathcal{U}(\mathbb{F}_2^k)}\left[\sum_{i=1}^k y_i x_i = 1\right] = \frac{1}{2}.$$

If $F_\ell(\alpha) = 1$ then $\Pr\left[\sum_{j=1}^t e_j f_j(\alpha) \neq 0\right] = \frac{1}{2}$ by the random subsum principle. If $F_\ell(\alpha) = 0$, then $\Pr\left[\sum_{j=1}^t e_j f_j(\alpha) = 0\right] = 1$. ◀

▶ Remark 14. Similarly to the proof of Lemma 1 one can prove that if an unsatisfiable CNF formula φ has a tree-like $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ refutation of length ℓ, then for any k-partition of the variables, there is a randomized bounded-error k-party NOF protocol for Search(φ) with communication cost $\mathcal{O}(c \log \ell)$. Thus, the bound from Lemma 9 can be slightly improved in the case of one-sided error.

4 Perfect matching

In this section we prove the following theorem:

▶ **Theorem 2.** *The size of any tree-like semantic* $\mathrm{Res}(\oplus)$ *refutation of the formula* $\mathrm{PM}_{K_{n+2,n}}$ *is* $2^{\Omega(n)}$.

By Lemma 1, to prove Theorem 2 it is sufficient to show that $R^{1/3}_{pub}\left(\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)\right) = \Omega(n)$.

Consider the communication problem $\oplus\mathrm{PM}_n^m$ that is defined as follows: Alice and Bob have matrices X and Y over $\mathbb{F}_2$ respectively, each of the matrices has size $m \times n$, where $m \neq n$. Their goal is to find an all-zero row or column or two 1-cells in the same row or column in the matrix $X + Y$.

▶ **Proposition 15.** $R^{1/3}_{pub}\left(\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)\right) \ge R^{1/3}_{pub}(\oplus\mathrm{PM}_n^{n+2})$.

Proof. A Boolean matrix of size $(n+2) \times n$ naturally corresponds to a subset of edges of $K_{n+2,n}$. A falsified clause encoding that a vertex must be covered by a matching corresponds to an all-zero row or column of the matrix; a falsified clause, encoding that a vertex can not be covered by a matching twice, corresponds to two ones in the same row or column. ◀

Theorem 2 follows from Proposition 15 and the following theorem.

▶ **Theorem 16.** $R^{1/3}_{pub}(\oplus\mathrm{PM}^{n+2}_{n}) = \Omega(n)$.

Proof. We assume that $n = 4m + 1$, where m is a non-negative integer. If the theorem is true for all n with the residue 1 modulo 4, then it also holds for all other n. Indeed, the protocol for $\oplus\mathrm{PM}^{n+3}_{n+1}$ can be used for $\oplus\mathrm{PM}^{n+2}_{n}$ by adding to Alice's matrix an extra column and a row with exactly one 1-cell on their intersection and to Bob's matrix an extra column and a row with all zeros.

Let $\mathcal{P}_0$ be a protocol for $\oplus\mathrm{PM}^{n+2}_{n}$ transmitting at most k bits. We are going to apply $\mathcal{P}_0(X, Y)$ only to the instances where the matrix $X + Y$ does not contain all-zero rows or columns. Thus, we assume that with probability at least $2/3$ $\mathcal{P}_0$ returns a tuple $(r_1, c_1, r_2, c_2) \in ([n + 2] \times [n])^2$ such that $(X + Y)_{r_1,c_1} = (X + Y)_{r_2,c_2} = 1$ and either $\begin{cases} r_1 = r_2 \\ c_1 \neq c_2 \end{cases}$ or $\begin{cases} r_1 \neq r_2 \\ c_1 = c_2 \end{cases}$. With $\mathcal{O}(1)$ bits of communication Alice and Bob can verify whether the answer of $\mathcal{P}_0$ is correct and return $\perp$ (failure) if it is not. Also, we can reduce the failure probability by the repetition of the protocol. Let $\mathcal{P}$ be a protocol for $\oplus\mathrm{PM}^{n+2}_{n}$ under the promise that $X + Y$ does not contain all-zero rows and columns that uses $\mathcal{O}(k)$ bits of communication and returns a correct answer with probability at least $\frac{99}{100}$ and $\perp$ otherwise.

We are going to construct a protocol for DISJ_m transmitting $\mathcal{O}(k)$ bits, where $m = \frac{n-1}{4}$. Since by Theorem 8 any protocol for DISJ_m transmits $\Omega(m)$ bits, we conclude that $k = \Omega(m)$. Let Alice's input for DISJ_m be $a_1, \dots, a_m$ and Bob's input be $b_1, \dots, b_m$.

▶ **Lemma 17.** *There exist matrices $A(0), A(1), B(0), B(1) \in \mathbb{F}_2^{4\times 4}$ such that $A(x) + B(y)$ is a permutation matrix iff $x \wedge y$ is 0 and*

$$A(1) + B(1) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

Proof. We simply present matrices that satisfy the conditions:

$$A(0) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};\ A(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$B(0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix};\ B(1) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$ ◀

Notice that Lemma 17 immediately allows to reduce DISJ_m to the problem of *checking* whether the sum of Alices and Bobs matrices is a permutation matrix. In order to achieve that, Alice builds a matrix $\mathcal{A} = \mathrm{diag}(A(a_1), \dots, A(a_m))$, Bob builds a matrix $\mathcal{B} = \mathrm{diag}(B(b_1), \dots, B(b_m))$. It is easy to see that $\mathcal{A} + \mathcal{B}$ is a permutation matrix iff $\mathrm{DISJ}_m(a, b) = 1$.

Let us describe the reduction of DISJ_m to $\oplus\mathrm{PM}_n^{n+2}$. Alice and Bob first construct matrices X_0 and Y_0 of the following form:

$$X_0 = \begin{pmatrix} \mathcal{A} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \end{pmatrix}; \quad Y_0 = \begin{pmatrix} \mathcal{B} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 0 \\ \mathbf{0}_{1\times(n-1)} & 0 \\ \mathbf{0}_{1\times(n-1)} & 0 \end{pmatrix},$$

then

$$X_0 + Y_0 = \begin{pmatrix} \mathcal{A}+\mathcal{B} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \end{pmatrix},$$

where $\mathcal{A}+\mathcal{B}$ is a permutation matrix iff $\mathrm{DISJ}_m(a,b)=1$. Then if $\mathcal{P}(X_0,Y_0)$ returns two cells that do not belong to the column n we may conclude that $\mathrm{DISJ}_m(a,b)=0$. If $\mathcal{P}(X_0,Y_0)$ returns two cells from the nth column, then the value of $\mathrm{DISJ}_m(a,b)$ can not be uniquely determined. Notice that for X_0 and Y_0 constructed as above the protocol always returning $(n+1,n,n+2,n)$ solves $\oplus\mathrm{PM}_n^{n+2}$.

If $\mathrm{DISJ}_m(a,b)=0$, then the matrix X_0+Y_0 contains at least two columns with three ones and these columns are indistinguishable from each over. To make use of that, we randomly shuffle rows and columns.

We are going to construct a protocol $\mathcal{T}$ for DISJ_m as follows: Alice and Bob choose permutations $\pi\in S_n$, $\tau\in S_{n+2}$ and a matrix $\Delta\in\mathbb{F}_2^{(n+2)\times n}$ uniformly at random. We define matrices $X_0^{\tau,\pi}$ and $Y_0^{\tau,\pi}$ from $\mathbb{F}_2^{n+2\times n}$ such that for each $i\in[n+2]$ and $j\in[n]$, $(X_0^{\tau,\pi})_{i,j}=(X_0)_{\tau(i),\pi(j)}$ and $(Y_0^{\tau,\pi})_{i,j}=(Y_0)_{\tau(i),\pi(j)}$. Alice and Bob run the protocol $\mathcal{P}$ for inputs $X=X_0^{\tau,\pi}+\Delta$, $Y=Y_0^{\tau,\pi}+\Delta$. Notice that $X+Y=X_0^{\tau,\pi}+Y_0^{\tau,\pi}$, thus $X+Y$ can be obtained from X_0+Y_0 by shuffling rows and columns. If $\mathcal{P}(X,Y)$ returns two cells from the column $\pi(n)$, Alice and Bob return 1, if $\mathcal{P}(X,Y)$ returns two cells from other column or row, Alice and Bob return 0. If $\mathcal{P}(X,Y)$ returns $\perp$, then Alice and Bob return $\perp$.

First notice that if $\mathrm{DISJ}_m(a,b)=1$, then $\mathcal{T}$ returns a correct answer or $\perp$ with probability 1 (and the probability of $\perp$ is at most $\frac{1}{100}$), since in that case $X+Y$ has exactly one column with three 1-cells, each of the other columns and rows contains exactly one 1-cell. Let us fix $a,b\in\{0,1\}^m$ such that $\mathrm{DISJ}_m(a,b)=0$. We denote $p:=\Pr[\mathcal{T}(a,b)=0]$, we will show that $p\geq\frac{99}{200}$. We can then increase this probability to $2/3$ by repeating the protocol twice (if $\mathcal{T}(a,b)$ returns 0 at least once, we return 0, if $\mathcal{T}(a,b)$ always return $\perp$, we return $\perp$, otherwise we return 1).

Let us describe random bits used by the constructed protocol $\mathcal{T}$. First, we use random bits r to run the protocol $\mathcal{P}$. Second, we use random bits to generate π,τ, and Δ. Since $\mathrm{DISJ}_m(a,b)=0$, we can fix $i\in[m]$ such that $a_i=b_i=1$. In that case the submatrix of X_0+Y_0 formed by rows and columns with the indices $4(i-1)+1, 4(i-1)+2, 4(i-1)+3, 4(i-1)+4$ coincides with the matrix (1). Let us denote by $\mathrm{col}(j)$ for $j\in[n]$ the set of all tuples $(x,j,y,j)\in([n+2]\times[n])^2$.

$$\begin{aligned}
p &= \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y)\notin\mathrm{col}(\pi(n))] - \overbrace{\Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y)=\perp]}^{=:p_\perp} \\
&= 1 - \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y)\in\mathrm{col}(\pi(n))] - p_\perp \\
&= 1 - \sum_{\pi_0,\tau_0}\Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)\in\mathrm{col}(\pi_0(n))]\Pr_{\pi,\tau}[\pi=\pi_0,\tau=\tau_0] - p_\perp
\end{aligned}$$

Observe that for fixed π_0 and τ_0 the random variable $(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)$ is uniformly distributed over the pairs of matrices with the sum $X_0^{\tau_0,\pi_0}+Y_0^{\tau_0,\pi_0}$. Let $\alpha \in S_n$ be the transposition swapping n and $4(i-1)+1$. Let $\beta \in S_{n+2}$ be the permutation swapping n and $4(i-1)+2$, $n+1$ and $4(i-1)+3$, $n+2$ and $4(i-1)+3$ (i.e. β is a product of three transpositions). By the construction of α and β, $(X_0+Y_0)=(X_0^{\beta,\alpha}+Y_0^{\beta,\alpha})$, thus $(X_0^{\tau,\pi}+Y_0^{\tau,\pi})=(X_0^{\tau\circ\beta,\pi\circ\alpha}+Y_0^{\tau\circ\beta,\pi\circ\alpha})$ for every π,τ. Thus the random variable $(X_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta, Y_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta)$ has the same distribution with $(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)$, thus we can continue the sequence as follows:

$$\begin{aligned}
p &= 1-\sum_{\pi_0,\tau_0}\Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta, Y_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta)\in \mathrm{col}(\pi_0(n))]\Pr_{\pi,\tau}[\pi=\pi_0,\tau=\tau_0]-p_\perp\\
&= 1-\sum_{\pi_0,\tau_0}\Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)\in \mathrm{col}(\pi_0\circ\alpha^{-1}(n))]\Pr_{\pi,\tau}[\pi=\pi_0,\tau=\tau_0]-p_\perp\\
&= 1-\Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y)\in \mathrm{col}((\pi\circ\alpha^{-1})(n))]-p_\perp\\
&\geq 1-\Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y)\notin \mathrm{col}(\pi(n))]-p_\perp = 1-p-p_\perp
\end{aligned}$$

Thus, $p\geq 1-p-p_\perp$ and $p\geq \frac{1-p_\perp}{2}=\frac{99}{200}$. ◀

5 Bit pigeonhole principle with parity gadget

In this section, we prove the following theorem.

▶ **Theorem 3.** *Let ℓ and k be natural numbers such that $2\leq k\leq \ell-7$. Then*

$$R_{pub}^{1/3}\left(\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)\right)=\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right).$$

For $k=2$ the stronger bound holds: $R_{pub}^{1/3}\left(\oplus_2\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+4}\right)\right)=\Omega\left(2^\ell\right)$.

We consider a combinatorial analogue of the communication problem $\oplus_k\mathrm{Search}\,(\mathrm{BPHP}_{2^\ell}^m)$. Assume that each of k parties gets m binary strings from $\{0,1\}^\ell$, where $m>2^\ell$. The ith party has numbers $a_{i,1},\ldots,a_{i,m}\in\{0,1\}^\ell$ on their forehead. Based on these strings we form the following set of m vectors from $\mathbb{F}_2^\ell$: $x_1,x_2,\ldots,x_m$, where $x_j=\sum_{i=1}^k a_{i,j}$. The goal of the parties is to find a pair of different indices $t,s\in[m]$ such that $x_t=x_s$. We denote this problem by $\oplus_k\mathrm{BPHP}_{2^\ell}^m$. It is straightforward that $R_{pub}^{1/3}\left(\oplus_k\mathrm{Search}\,(\mathrm{BPHP}_{2^\ell}^m)\right)\geq R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^m\right)$, hence it is sufficient to prove a lower bound on $R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^m\right)$.

▶ **Theorem 18.** *Let ℓ and k be natural numbers such that $2\leq k\leq \ell-7$. Then*

$$R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)=\Omega\left(R_{pub}^{1/3}\left(\mathrm{UDISJ}_{k,2^{\ell-k}-1}\right)-\ell\right).$$

▶ **Corollary 19.** $R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)=\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)$. *For $k=2$ the stronger bound holds:* $R_{pub}^{1/3}\left(\oplus_2\mathrm{BPHP}_{2^\ell}^{2^\ell+4}\right)=\Omega\left(2^\ell\right)$.

Proof of Corollary 19 . Follows from Theorem 18 and Theorem 7; for $k=2$ we should apply Theorem 8. ◀

Theorem 3 immediately follows from Corollary 19.

5.1 Warm-up example

We start with the simpler statement that, nonetheless, demonstrates the main idea of Theorem 18. Consider the following communication problem $\text{Distinct}_{k,\ell}$: let each of k parties have a matrix from $\mathbb{F}_2^{2^\ell \times \ell}$ on their forehead. The goal is to determine whether all rows of the sum of all these matrices are distinct. A version of this problem without the xor-gadget is referred to as *Element Distinctness* (ED) in the literature [25].

▶ **Proposition 20.** $R^{1/3}_{pub}(\text{Distinct}_{k,\ell}) \geq R^{1/3}_{pub}\left(\text{UDISJ}_{k,2^{\ell-k}}\right)$.

Let $\mathbb{S}_k$ denote the set of matrices from $\{0,1\}^{2^k \times k}$ with all distinct rows. Let $K_k \in \{0,1\}^{2^k \times k}$ be a matrix such that its ith row equals $\texttt{bin}_k(i-1-((i-1) \bmod 2))$, i.e. the rows of K_k are $\texttt{bin}_k(0), \texttt{bin}_k(0), \texttt{bin}_k(2), \texttt{bin}_k(2), \ldots, \texttt{bin}_k(2^{k-1}-2), \texttt{bin}_k(2^{k-1}-2)$. Notice that every row of K_k starts with zero and appears exactly twice.

In the proof of Proposition 20 as well as in the proof of Theorem 18 we will use the following combinatorial lemma that we prove in Subsection 5.4.

▶ **Lemma 21.** *There exist matrices $A_1(0), A_1(1), \ldots, A_k(0), A_k(1) \in \mathbb{F}_2^{2^k \times k}$ such that $\sum_{i=1}^k A_i(1) = K_k$ and for all $b_1, b_2 \ldots, b_k \in \{0,1\}$, if $\bigwedge_{i=1}^k b_i = 0$, then $\sum_{i=1}^k A_i(b_i) \in \mathbb{S}_k$.*

Proof of Proposition 20. Let $(x_{i,1}, \ldots, x_{i,2^{\ell-k}})$ be an input of the ith party of the problem $\text{UDISJ}_{k,2^{\ell-k}}$. For all $i \in [k]$ we construct a matrix D_i of size $2^\ell \times \ell$ and put it on the forehead of the ith party. Let $A_i(b)$ for $i \in [k]$, $b \in \{0,1\}$ be matrices of size $2^k \times k$ from Lemma 21. Let J_t for $t \in [1, \ldots, 2^{\ell-k}]$ be a matrix of size $2^k \times (\ell - k)$ such that all its rows are equal to $\texttt{bin}_{\ell-k}(t-1)$.

Let us define

$$D_1 := \begin{pmatrix} J_1 & A_1(x_{1,1}) \\ \vdots & \vdots \\ J_j & A_1(x_{1,j}) \\ \vdots & \vdots \\ J_{2^{\ell-k}} & A_1(x_{1,2^{\ell-k}}) \end{pmatrix}; \quad D_i := \begin{pmatrix} \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,1}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,j}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,2^{\ell-k}}) \end{pmatrix} \text{ for } i \in \{2, \ldots, k\}.$$

By Lemma 21, the matrix $D_1 + D_2 + \cdots + D_k$ has the following property: for all $j \in [2^{\ell-k}]$, its submatrix formed by the rows with numbers from $[2^k \cdot (j-1)+1, 2^k \cdot j]$ has two equal rows if and only if $x_{1,j} = x_{2,j} = \ldots = x_{k,j} = 1$. Thus, the communication complexity of $\text{UDISJ}_{k,2^{\ell-k}}$ is at most the communication complexity of $\text{Distinct}_{k,\ell}$. ◀

5.2 Proof of Theorem 18

In order to prove Theorem 18 we modify the proof of Proposition 20 in order to reduce $\text{UDISJ}_{k,2^{\ell-k}-1}$ to $\oplus_k \text{BPHP}^{2^\ell+2^k}_{2^\ell}$ by adding "fake" rows (such rows do not correspond to the input of the unique disjointness) to matrices $D_1, D_2, \ldots, D_k$. We also use some randomization in order to hide "fake" rows among other rows.

Proof of Theorem 18. Let $N > 2^\ell$, consider a k-party communication problem $\text{ROW} \oplus_k \text{BPHP}^N_{2^\ell}$, where ith party has a matrix $M_i \in \mathbb{F}_2^{N \times \ell}$ on their forehead and their goal is to find the value of a row of $M_1 + \cdots + M_k$ that appears in this matrix at least twice. The difference with the problem $\oplus_k \text{BPHP}^N_{2^\ell}$ is that we are looking for values of a repeated row rather than numbers of equal rows.

▷ Claim 22. If $R_{1/3}\left(\oplus_k \mathrm{BPHP}^N_{2^\ell}\right) \le t$, then there exists a communication protocol $\mathcal{P}$ for $\mathrm{ROW} \oplus_k \mathrm{BPHP}^N_{2^\ell}$ using $\mathcal{O}(t+\ell)$ bits of communication such that $\mathcal{P}$ either returns the correct answer or $\perp$ (failure) and $\Pr[\mathcal{P}(M_1, \dots, M_k) = \perp] \le \frac{1}{100}$ for all input matrices M_i, $i \in [k]$.

Proof. $\mathcal{P}$ executes a randomized protocol for $\oplus_k \mathrm{BPHP}^N_{2^\ell}$ and verifies its answer by transferring additional $\mathcal{O}(\ell)$ bits. The probability of failure can be reduced by repetition. ◁

Let us describe a protocol for the problem $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ that uses a protocol $\mathcal{P}$ for $\mathrm{ROW} \oplus_k \mathrm{BPHP}^{2^\ell+2^k}_{2^\ell}$ from Claim 22.

Let $x_1, \dots, x_k \in \{0,1\}^{2^{\ell-k}-1}$ be inputs of the communication problem $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Let $x_{i,j}$ denote the jth bit of x_i for $i \in [k], j \in [2^{\ell-k}-1]$. Let $\overrightarrow{x} = (x_1, x_2, \dots, x_k)$.

Important matrices

Let γ be a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$, we define k matrices $D_1(x_1,\gamma)$ and $D_2(x_2), D_3(x_3), \dots, D_k(x_k)$ of size $(2^\ell + 2^k) \times \ell$ similar to Proposition 20.

Let $A_i(b)$ for $i \in [k]$, $b \in \{0,1\}$ be matrices of size $2^k \times k$ from Lemma 21. Let for every $t \in \{0,1\}^{\ell-k}$, J_t be a matrix of size $2^k \times (\ell - k)$ such that all its rows are equal to t. Let W be some fixed matrix from $\mathbb{S}_k$.

We define

$$D_1(x_1,\gamma) := \begin{pmatrix} J_{\gamma(1)} & A_1(x_{1,1}) \\ \vdots & \vdots \\ J_{\gamma(j)} & A_1(x_{1,j}) \\ \vdots & \vdots \\ J_{\gamma(2^{\ell-k}-1)} & A_1(x_{1,2^{\ell-k}-1}) \\ J_{\gamma(*)} & W \\ J_{\gamma(*)} & W \end{pmatrix};$$

and for $i \in [k] \setminus \{1\}$

$$D_i(x_i) := \begin{pmatrix} \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,1}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,j}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,2^{\ell-k}-1}) \\ \mathbf{0}_{2^k \times (\ell-k)} & \mathbf{0}_{2^k \times k} \\ \mathbf{0}_{2^k \times (\ell-k)} & \mathbf{0}_{2^k \times k} \end{pmatrix}.$$

Notice that the submatrix of $D_1(x_1,\gamma)$ formed by the last 2^{k+1} rows of the matrix $D_1(x_1,\gamma)$ contains every its row exactly two times.

We define $H_{\overrightarrow{x}}(\gamma) := D_1(x_1,\gamma) + D_2(x_2) + \dots + D(x_k)$. By Lemma 21 the matrix $H_{\overrightarrow{x}}(\gamma)$ satisfies the following *key* property w.r.t. $(\gamma, \overrightarrow{x})$ in the standard basis:

▶ **Definition 23.** *Let M be a matrix from $\mathbb{F}_2^{(2^k+2^\ell)\times \ell}$, γ be a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and $e_1, e_2, \dots, e_\ell$ be a basis in $\mathbb{F}_\ell$.*

We say that M satisfies the key *property w.r.t $(\gamma, \overrightarrow{x})$ in the basis $(e_1, e_2, \dots, e_\ell)$ if the following properties hold:*

- *If s is a row among the last 2^{k+1} rows of M, then*
  - *the first $\ell - k$ coordinates of s in the basis $(e_1, e_2, \dots, e_\ell)$ are $\gamma(*)_1, \dots \gamma(*)_{\ell-k}$;*
  - *s appears in M exactly twice.*
- *If s is a row of M among the rows with numbers $[2^k(i-1)+1; 2^k i]$ for $i \in [2^{\ell-k}-1]$, then*
  - *the first $\ell - k$ coordinates of s in the basis $(e_1, e_2, \dots, e_\ell)$ are $\gamma(i)_1, \dots, \gamma(i)_{\ell-k}$;*
  - *if $\bigwedge_{j=1}^{k} x_{i,j} = 0$, then s appears in M exactly once.*
  - *if $\bigwedge_{j=1}^{k} x_{i,j} = 1$, then s appears in M exactly twice and $(\ell - k + 1)$th coordinate of s in the basis $(e_1, e_2, \dots, e_\ell)$ is 0.*

Consider an invertible matrix $E \in \mathbb{F}_2^{\ell \times \ell}$. Let $e_1, e_2, \dots, e_\ell$ be the rows of E. Since E is invertible, $e_1, e_2, \dots, e_\ell$ form a basis. Let us define $C_{\overrightarrow{x}}(\gamma, E) := H(\overrightarrow{x}, \gamma)E$. Rows of $C_{\overrightarrow{x}}(\gamma, E)$ can be viewed as vectors with coordinates in the basis $e_1, e_2, \dots, e_\ell$ corresponding to the rows of $H(\overrightarrow{x}, \gamma)$. Hence, $C_{\overrightarrow{x}}(\gamma, E)$ satisfies the key property w.r.t. $(\gamma, \overrightarrow{x})$ in the basis $(e_1, e_2, \dots, e_\ell)$.

For a bijection γ from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and an invertible matrix $E \in \mathbb{F}_2^{\ell \times \ell}$ we define a set $\mathrm{Fake}(\gamma, E) \subseteq \mathbb{F}_2^{\ell}$ as a set of the last 2^{k+1} rows of the matrix $C_{\overrightarrow{x}}(\gamma, E)$. Notice that by the construction this set does not depend on $\overrightarrow{x}$. By the key property rows from $\mathrm{Fake}(\gamma, E)$ appear exactly twice in $C_{\overrightarrow{x}}(\gamma, E)$.

Random variables

Our protocol uses the following public random variables. In order to distinguish random variables from their values, we highlight random variables in bold.

- $\boldsymbol{\gamma}$ is a random bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ distributed uniformly among all such bijections.
- $\boldsymbol{E}$ is a random invertible matrix from $\mathbb{F}_2^{\ell \times \ell}$ distributed uniformly among all such matrices.
- $\boldsymbol{\pi}$ is a random permutation of the set $[2^\ell + 2^k]$ and $M_{\boldsymbol{\pi}}$ is a permutation matrix of size $(2^\ell + 2^k) \times (2^\ell + 2^k)$ corresponding to the permutation $\boldsymbol{\pi}$ (i.e. $(M_{\boldsymbol{\pi}})_{i,j} = 1 \iff \boldsymbol{\pi}(i) = j$).
- $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \dots, \boldsymbol{\Delta}_k$ are random matrices from $\mathbb{F}_2^{(2^\ell + 2^k) \times \ell}$ distributed uniformly on the set of all matrices $\Delta_1, \Delta_2, \dots, \Delta_k$ such that $\Delta_1 + \Delta_2 + \ldots + \Delta_k$ is the zero matrix.

We define random matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_k$ as follows: $\boldsymbol{P}_i = M_{\boldsymbol{\pi}} \cdot D_i(x_i) \cdot \boldsymbol{E} + \boldsymbol{\Delta}_i$ for $i \geq 2$ and $\boldsymbol{P}_1 = M_{\boldsymbol{\pi}} \cdot D_1(x_1, \boldsymbol{\gamma}) \cdot \boldsymbol{E} + \boldsymbol{\Delta}_1$.

- The addition of $\boldsymbol{\Delta}_i$ makes $\boldsymbol{P}_i$ indistinguishable from the random matrix for every $i \in [k]$.
- $\sum_{i=1}^{k} \boldsymbol{P}_i = M_{\boldsymbol{\pi}} C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ and this matrix is obtained from $C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ by the permutation $\boldsymbol{\pi}$ applied to its rows.

Recall that $\mathcal{P}$ is the protocol for $\mathrm{ROW} \oplus_k \mathrm{BPHP}_{2^\ell}^{2^\ell + 2^k}$ from Claim 22. Let N be a constant to be chosen later. The protocol $\mathcal{T}$ solving $\mathrm{UDISJ}_{k, 2^{\ell-k}-1}$ is described by Algorithm 1.

Protocol analysis

Let us analyze the protocol $\mathcal{T}$. Since it executes the protocol $\mathcal{P}$ a constant number of times, $\mathcal{T}$ transmits $\mathcal{O}(t + \ell)$ bits. Assume that $x_1, x_2, \dots, x_k$ is a 1-instance of $\mathrm{UDISJ}_{k, 2^\ell + 2^k}$. Then by the key property of $C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ all repeated rows of $\sum_{i=1}^{k} \boldsymbol{P}_i$ are in $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$, hence the protocol $\mathcal{T}$ returns either $\perp$ or the correct answer. Since $\mathcal{P}$ is executed N times independently, the probability that $Z = \{\perp\}$ is at most $\frac{1}{100^N}$, hence $\mathcal{T}$ returns 1 with probability at least $1 - \frac{1}{100^N}$.

The rest of the proof is devoted to the analysis of the case, where $x_1, x_2, \dots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k, 2^\ell + 2^k}$. This is the most technically involved part of the proof. So it is a good point to give a **large scale overview of the further proof strategy**. Our goal

Algorithm 1 Protocol $\mathcal{T}$ solving $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$.

Input $x_1, x_2, \ldots, x_k \in \{0,1\}^{2^{\ell-k}-1}$; x_i is written on the forehead of the ith party for every $i \in [k]$.

$Z := \varnothing$
loop repeat N times
  Sample $\pi \leftarrow \boldsymbol{\pi}$, $E \leftarrow \boldsymbol{E}$, $\gamma \leftarrow \boldsymbol{\gamma}$, $\vec{\Delta} \leftarrow \vec{\boldsymbol{\Delta}}$ ▷ Use fresh public random bits
  $P_1 := M_\pi \cdot D_1(x_1, \gamma) \cdot E + \Delta_1$ ▷ Can be computed by parties $2, 3, \ldots, k$
  $P_i := M_\pi \cdot D_i(x_i) \cdot E + \Delta_i$ for $i \geq 2$ ▷ Can be computed by all parties except the ith
  $z := \mathcal{P}(P_1, \ldots, P_k)$ ▷ Use fresh random bits for $\mathcal{P}$ and assume that P_i is written on the ith party's forehead.
  $Z := Z \cup \{z\}$
if $Z = \{\perp\}$ **then return** $\perp$
else if $Z \setminus \{\perp\} \subseteq \mathrm{Fake}(\gamma, E)$ **then return** 1 ▷ Intuitively this step means that most likely there are no more repeated rows in $C_{\vec{x}}(\gamma, E)$ except $\mathrm{Fake}(\gamma, E)$ and, hence, $\mathrm{DISJ}(x_1, x_2, \ldots, x_k) = 1$ by the key property of $C_{\vec{x}}(\gamma, E)$.
return 0

is to show that if $x_1, x_2, \ldots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k,2^\ell+2^k}$, then the probability that $\mathcal{P}(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k)$ returns a value from $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$ is bounded by some constant less than 1. The random variable $\mathcal{P}(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k)$ depends on random bits used by the protocol $\mathcal{P}$ and on random bits needed for sampling $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k$. Let R denote the set of all random strings used by the protocol $\mathcal{P}$ (i.e. we assume that $\mathcal{P}$ sample a random string from R and use it as public randomness) and S denote the set of all random strings used for sampling $\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k$. We would like to construct two bijections α and β on the set S such that for every $s \in S$ the following two properties hold.

1. The three values of random variable $(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_k)$ sampled using three strings $s, \alpha(s)$ and $\beta(s)$ as a random source, are the same.
2. Let $(\gamma, E), (\gamma_\alpha, E_\alpha)$ and (γ_β, E_β) be values of the random variable $(\boldsymbol{\gamma}, \boldsymbol{E})$ that is sampled using three strings $s, \alpha(s)$ and $\beta(s)$ as a random source. Then $\mathrm{Fake}(\gamma, E) \cap \mathrm{Fake}(\gamma_\alpha, E_\alpha) \cap \mathrm{Fake}(\gamma_\beta, E_\beta) = \varnothing$.

Consider arbitrary strings $r \in R$ and $s \in S$. The first property implies that for random variables sampled using strings (r, s), $(r, \alpha(s))$ and $(r, \beta(s))$ as a random source values of $\mathcal{P}(\boldsymbol{P_1}, \ldots, \boldsymbol{P_k})$ are the same. The second property implies that for at least one of this cases this value does not belong to $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$. Then, using that α and β are bijections, we get $\Pr[\mathcal{P}(\boldsymbol{P_1}, \ldots, \boldsymbol{P_k}) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})] \leq \frac{2}{3}$.

Since we have many random variables, it is a tedious task to construct such α and β. In order to simplify this task we slightly relax the properties. We will define bijections α and β not on all strings S but only on the part of bits corresponding to sampling of $\boldsymbol{\gamma}$ and $\boldsymbol{E}$. More precisely we will define two bijections α and β on the set of values of the random variable $(\boldsymbol{\gamma}, \boldsymbol{E})$. We relax the first property as follows:

1'. For every γ and E the three conditional distributions of the random variable $(\boldsymbol{P_1}, \ldots, \boldsymbol{P_k})$ under the following three conditions coincide:

a. $(\boldsymbol{\gamma}, \boldsymbol{E}) = (\gamma, E)$,
b. $(\boldsymbol{\gamma}, \boldsymbol{E}) = \alpha(\gamma, E)$ and
c. $(\boldsymbol{\gamma}, \boldsymbol{E}) = \beta(\gamma, E)$.

Unfortunately, we were not able to construct such bijections on the set of all pairs (γ, E). Thus we take a set Ξ consisting $1-\delta$ fraction of all values of $(\boldsymbol{\gamma}, \boldsymbol{E})$ and we will claim that α and β are bijections on Ξ. Such relaxation will weaken the bound of the probability up to $\frac{2}{3}+\delta$. We formalize the requirements to Ξ, α and β in Definition 24. Then we verify in Claim 25 that these requirements are sufficient to bound $\Pr[\mathcal{P}(\boldsymbol{P_1}, \ldots, \boldsymbol{P_k}) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})]$. The construction of Ξ, α and β is given in Subsection 5.3.

▶ **Definition 24.** *Let $x_1, \ldots, x_k$ be a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ and $1 > \delta \geq 0$ be an arbitrary constant. Let Ξ be a set consisting of pairs (γ, E), where γ is a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$, E is an invertible matrix from $\mathbb{F}_2^{\ell\times\ell}$. Let α and β be bijections from Ξ to Ξ. We say that (Ξ, α, β) forms a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$ if the following conditions hold:*

- *(Largeness)* $\Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi] \geq 1-\delta$.
- *(Difference) For all $(\gamma, E) \in \Xi$,* $\mathrm{Fake}(\gamma, E) \cap \mathrm{Fake}(\alpha(\gamma, E)) \cap \mathrm{Fake}(\beta(\gamma, E)) = \varnothing$.
- *(Symmetry) For all $(\gamma, E) \in \Xi$ the matrices $C_{\overrightarrow{x}}(\gamma, E)$, $C_{\overrightarrow{x}}(\alpha(\gamma, E))$ and $C_{\overrightarrow{x}}(\beta(\gamma, E))$ differ only by a permutation of rows.*

▷ Claim 25. Assume that $x_1, \ldots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$, $1 > \delta \geq 0$ is a constant. Let (Ξ, α, β) form a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$

Then

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_k\right) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})\right] \leq \frac{2}{3} + \delta.$$

Proof. Let us denote $\overrightarrow{\boldsymbol{P}} = (\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_k)$, $\overrightarrow{\boldsymbol{\Delta}} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \ldots, \boldsymbol{\Delta}_k)$ and $\overrightarrow{D}(\overrightarrow{x}, \gamma) = (D_1(x_1, \gamma), D_2(x_2), \ldots, D_k(x_k))$.

$\overrightarrow{\boldsymbol{P}} = (\boldsymbol{\Delta}_1 + M_{\boldsymbol{\pi}} D_1(x_1, \boldsymbol{\gamma})\boldsymbol{E}, \boldsymbol{\Delta}_2 + M_{\boldsymbol{\pi}} D_2(x_2)\boldsymbol{E}, \ldots, \boldsymbol{\Delta}_k + M_{\boldsymbol{\pi}} D_k(x_k)\boldsymbol{E})$, for brevity we use the vector notation $\overrightarrow{\boldsymbol{P}} = \overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}(\overrightarrow{D}(\overrightarrow{x}, \boldsymbol{\gamma})\boldsymbol{E})$.

Let $p := \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})\right]$.

$$p = \sum_{\gamma, E} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\left(\overrightarrow{D}(\overrightarrow{x}, \gamma)\cdot E\right)\right) \in \mathrm{Fake}(\gamma, E)\right] \cdot \Pr[\boldsymbol{\gamma} = \gamma, \boldsymbol{E} = E]$$
$$\overset{\text{(Largeness)}}{\leq} \sum_{(\gamma, E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x}, \gamma)\cdot E\right)\right) \in \mathrm{Fake}(\gamma, E)\right] \cdot \Pr[\boldsymbol{\gamma} = \gamma, \boldsymbol{E} = E] + \delta$$

Notice that for fixed γ, E the random variable $\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x}, \gamma)\cdot E\right)$ is distributed uniformly on the set of tuples $(L_1, \ldots, L_k)$ of k matrices from $\mathbb{F}_2^{(2^\ell+2^k)\times\ell}$ such that $\sum_{i=1}^k L_i$ differs from $C_{\overrightarrow{x}}(\gamma, E)$ only by a permutation of rows. Let $(\gamma_{\alpha^{-1}}, E_{\alpha^{-1}}) = \alpha^{-1}(\gamma, E)$. By the symmetry condition, matrices $C_{\overrightarrow{x}}(\gamma, E)$ and $C_{\overrightarrow{x}}(\gamma_{\alpha^{-1}}, E_{\alpha^{-1}})$ differ only by permutation of rows. Thus, for every set A the probability $\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x}, \gamma)\cdot E\right)\right) \in A\right] = \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x}, \gamma_{\alpha^{-1}})\cdot E_{\alpha^{-1}}\right)\right) \in A\right]$. Hence,

$$
\begin{aligned}
p \leq &\sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma_{\alpha^{-1}}) \cdot E_{\alpha^{-1}}\right)\right) \in \mathrm{Fake}(\gamma, E)\right] \cdot \Pr[\boldsymbol{\gamma} = \gamma, \boldsymbol{E} = E] + \delta \\
= &\sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)\right) \in \mathrm{Fake}(\alpha(\gamma, E))\right] \cdot \Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) = \alpha(\gamma, E)] + \delta \\
= &\sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)\right) \in \mathrm{Fake}(\alpha(\gamma, E))\right] \cdot \Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) = (\gamma, E)] + \delta \\
= &\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \in \mathrm{Fake}(\alpha(\boldsymbol{\gamma}, \boldsymbol{E})), (\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi\right] + \delta.
\end{aligned}
$$

Analogously, $p \leq \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \in \mathrm{Fake}(\beta(\boldsymbol{\gamma}, \boldsymbol{E})), (\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi\right] + \delta$. Also the inequality $p \leq \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E}), (\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi\right] + \delta$ follows by the largeness condition. Then,

$$
\begin{aligned}
3(1-p) \geq &\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \notin \mathrm{Fake}(\beta(\boldsymbol{\gamma}, \boldsymbol{E})) \vee (\boldsymbol{\gamma}, \boldsymbol{E}) \notin \Xi\right] \\
&+ \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \notin \mathrm{Fake}(\alpha(\boldsymbol{\gamma}, \boldsymbol{E})) \vee (\boldsymbol{\gamma}, \boldsymbol{E}) \notin \Xi\right] \\
&+ \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \notin \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E}) \vee (\boldsymbol{\gamma}, \boldsymbol{E}) \notin \Xi\right] - 3\delta \\
\geq &\Pr\left[\mathcal{P}\left(\overrightarrow{P}\right) \notin \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E}) \cap \mathrm{Fake}(\alpha(\boldsymbol{\gamma}, \boldsymbol{E})) \cap \mathrm{Fake}(\beta(\boldsymbol{\gamma}, \boldsymbol{E})) \vee (\boldsymbol{\gamma}, \boldsymbol{E}) \notin \Xi\right] - 3\delta \\
= &1 - 3\delta.
\end{aligned}
$$

The last equality follows by the difference condition. Hence, $3(1-p) \geq 1 - 3\delta$, thus $p \leq \frac{2}{3} + \delta$. ◁

We prove the following lemma in Subsection 5.3

▶ **Lemma 26.** *Let $x_1, \ldots, x_k$ be a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Then for some $\delta < \frac{1}{3} - \frac{1}{100}$ there exists a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$.*

Lemma 26 and Claim 25 imply that there is a constant $\varepsilon > 0$ such that

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_k\right) \notin \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})\right] \geq \varepsilon + \frac{1}{100}.$$

Thus,

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_k\right) \notin \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E}) \cup \{\bot\}\right] \geq \varepsilon.$$

Then, for $N = \mathcal{O}\left(\log \frac{1}{\varepsilon}\right)$, $\mathcal{T}$ gives a correct answer for every 0-instance with probability at least $\frac{2}{3}$. ◀

5.3 Constructions of Ξ, α and β

Proof of Lemma 26. Assume that $x_1, \ldots, x_k$ is a 0-instance $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Let $i_0 \in [2^{\ell-k}-1]$ be such that $x_{1,i_0} = x_{2,i_0} = \ldots = x_{k,i_0} = 1$.

Hereinafter γ denotes a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$, E denotes an invertible matrix from $\mathbb{F}_2^{\ell\times\ell}$ and $e_1, e_2, \ldots, e_\ell$ denote rows of E.

Before presenting constructions of Ξ, α, and β we explain how we are going to establish symmetry and difference properties from Definition 24.

For every $s \in \{0,1\}^{\ell-k}$ and $b \in \{0,1\}$ we introduce the following notation:

$$R(s, b, E) := \left\{(s, b, z) \cdot E \mid z \in \mathbb{F}_2^{k-1}\right\}.$$

Using the key property of the matrix $C_{\overrightarrow{x}}(\gamma, E)$ we can describe rows of $C_{\overrightarrow{x}}(\gamma, E)$ in terms of $R(s, b, E)$.

▷ Claim 27.

- The set of the last 2^{k+1} rows of $C_{\overrightarrow{x}}(\gamma, E)$ is $R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ and each of this rows appears exactly twice. Recall that we already denote this set as $\mathrm{Fake}(\gamma, E)$. Hence, $\mathrm{Fake}(\gamma, E) = R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$.
- The set of rows of $C_{\overrightarrow{x}}(\gamma, E)$ with indices from $[2^k(i-1)+1; 2^k i]$ for $i \in [2^{\ell-k-1}] \setminus \{i_0\}$ is exactly $R(\gamma(i), 0, E) \cup R(\gamma(i), 1, E)$ and every such row appears exactly once.
- The set of rows of $C_{\overrightarrow{x}}(\gamma, E)$ with indices from $[2^k(i_0-1)+1; 2^k i_0]$ is exactly $R(\gamma(i_0), 0, E)$ and every such row appears exactly twice.

▷ Claim 28. $R(s, b, E)$ can be represented as a shift of the linear space $\mathrm{Span}(e_{\ell-k+2}, \ldots, e_\ell)$:

$$R(s,b,E) = \left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \ldots, e_\ell).$$

Proof.

$$R(s,b,E) = \left\{(s,b,z)\cdot E \mid z \in \mathbb{F}_2^{k-1}\right\} = \left\{(s,b,z)\cdot(e_1, e_2, \ldots, e_\ell)^T \mid z \in \mathbb{F}_2^{k-1}\right\} =$$
$$\left\{\sum_{i=j}^{\ell-k} s_j e_j + b\cdot e_{\ell-k+1} + \sum_{i=1}^{k-1} z_i e_{\ell-k+1+i} \mid z \in \mathbb{F}_2^{k-1}\right\} =$$
$$\left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \ldots, e_\ell).$$

◁

▷ Claim 29. For every $s \in \{0,1\}^{\ell-k}$ and $b \in \{0,1\}$, $|R(s,b,E)| = 2^{k-1}$.

Proof. By Claim 28, $|R(s,b,E)| = \left|\left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \ldots, e_\ell)\right| = |\mathrm{Span}(e_{\ell-k+2}, \ldots, e_\ell)| = 2^{k-1}$. ◁

▷ Claim 30. Sets $R(s,b,E)$ for $s \in \{0,1\}^{\ell-k}$ and $b \in \{0,1\}$ are disjoint.

Proof. Consider two vectors $u \in R(s,b,E)$ and $v \in R(s',b',E)$ such that $(s,b) \neq (s',b')$. Then, by Claim 28, u and v have different coordinates in the basis $e_1, e_2, \ldots, e_\ell$, hence $u \neq v$. ◁

▷ Claim 31. Assume that γ, γ' are bijections from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and E and E' are invertible matrices from $\mathbb{F}_2^{\ell\times\ell}$ such that

- $R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E) = R(\gamma'(i_0), 0, E') \cup R(\gamma'(*), 0, E') \cup R(\gamma'(*), 1, E')$;
- $R(\gamma(i_0), 1, E) = R(\gamma'(i_0), 1, E')$.

Then matrices $C_{\overrightarrow{x}}(\gamma, E)$ and $C_{\overrightarrow{x}}(\gamma', E')$ differ only by a permutation of rows.

Proof. By Claim 27, rows from $R(\gamma(i_0), 1, E)$ do not appear in $C_{\overrightarrow{x}}(\gamma, E)$, rows from $R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ appear in $C_{\overrightarrow{x}}(\gamma, E)$ exactly twice. The matrix $C_{\overrightarrow{x}}(\gamma, E)$ has $2^\ell + 2^k$ rows. All rows of $C_{\overrightarrow{x}}(\gamma, E)$ that are not in $R(\gamma(i_0), 1, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$, by Claim 27, appear in $C_{\overrightarrow{x}}(\gamma, E)$ exactly once.

By Claims 29 and 30, $|R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)| = 3 \cdot 2^{k-1}$, hence, the number of rows of $C_{\overrightarrow{x}}(\gamma, E)$ that are not in $R(\gamma(i_0), 1, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ equals $2^\ell - 2^{k+1}$. By Claims 29 and 30, the number of ℓ-bit strings not from $R(\gamma(i_0), 1, E) \cup R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ is also $2^\ell - 2^{k+1}$. Hence, all rows from $\{0,1\}^\ell \setminus (R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup (\gamma(*), 1, E) \cup R(\gamma(i_0), 1, E))$ appear in $C_{\overrightarrow{x}}(\gamma, E)$ exactly once. Thus, matrices $C_{\overrightarrow{x}}(\gamma, E)$ and $C_{\overrightarrow{x}}(\gamma', E')$ have the same set of rows and each row appears the same number of times in each of these matrices. $\triangleleft$

For $\alpha, \beta : \Xi \to \Xi$ we denote $\alpha(\gamma, E) = (\gamma_\alpha, E_\alpha)$ and $\beta(\gamma, E) = (\gamma_\beta, E_\beta)$. We are going to construct α and β such that for all $(\gamma, E) \in \Xi$ the following equalities are satisfied.

$$\begin{cases} R(\gamma(i_0), 1, E) & = & R(\gamma_\alpha(i_0), 1, E_\alpha) & = & R(\gamma_\beta(i_0), 1, E_\beta); \\ R(\gamma(i_0), 0, E) & = & R(\gamma_\alpha(*), 0, E_\alpha) & = & R(\gamma_\beta(*), 0, E_\beta); \\ R(\gamma(*), 1, E) & = & R(\gamma_\alpha(*), 1, E_\alpha) & = & R(\gamma_\beta(i_0), 0, E_\beta); \\ R(\gamma(*), 0, E) & = & R(\gamma_\alpha(i_0), 0, E_\alpha) & = & R(\gamma_\beta(*), 1, E_\beta). \end{cases} \tag{2}$$

Notice that by Claim 31, equations (2) imply the symmetry property. Equations (2) also imply the difference property. Indeed,

- $\mathrm{Fake}(\gamma, E) = R(\gamma(*), 1, E) \cup R(\gamma(*), 0, E)$;
- $\mathrm{Fake}(\gamma_\alpha, E_\alpha) = R(\gamma_\alpha(*), 1, E_\alpha) \cup R(\gamma_\alpha(*), 0, E_\alpha) = R(\gamma(*), 1, E) \cup R(\gamma(i_0), 0, E)$;
- $\mathrm{Fake}(\gamma_\beta, E_\beta) = R(\gamma_\beta(*), 1, E_\beta) \cup R(\gamma_\beta(*), 0, E_\beta) = R(\gamma(*), 0, E) \cup R(\gamma(i_0), 0, E)$.

Hence, by Claim 30, $\mathrm{Fake}(\gamma, E) \cap \mathrm{Fake}(\gamma_\alpha, E_\alpha) \cap \mathrm{Fake}(\gamma_\beta, E_\beta) = \varnothing$.

In order to complete the proof of the lemma we have to construct Ξ and bijections α, β from Ξ to Ξ such that

- (Largeness) $\Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi] > \frac{2}{3} + \frac{1}{100}$;
- and for all $(\gamma, E) \in \Xi$ the equations (2) are satisfied.

Definition of Ξ. A pair (γ, E) is in Ξ iff there exist $m, n \in [\ell - k]$ such that $(\gamma(*))_m = 1, (\gamma(i_0))_m = 0$ and $(\gamma(*))_n = 0, (\gamma(i_0))_n = 1$. In other words, $\gamma(*)$ and $\gamma(i_0)$ are not comparable with respect to coordinate-wise comparison.

Notice that $\boldsymbol{\gamma}(i_0)$ and $\boldsymbol{\gamma}(*)$ are distributed uniformly among non-equal elements of $\{0,1\}^{\ell-k}$. Let $\boldsymbol{S}$ and $\boldsymbol{T}$ are two independent random variables distributed uniformly on the set of all subsets of $[\ell - k]$. Then,

$$\begin{aligned} \Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi] &= 1 - \Pr[\boldsymbol{\gamma}(i_0) \le \boldsymbol{\gamma}(*) \vee \boldsymbol{\gamma}(*) \le \boldsymbol{\gamma}(i_0)] \ge 1 - 2\Pr[\boldsymbol{\gamma}(i_0) \le \boldsymbol{\gamma}(*)] \\ &= 1 - 2\Pr[\boldsymbol{S} \subseteq \boldsymbol{T} \mid \boldsymbol{S} \ne \boldsymbol{T}] \ge 1 - 2\Pr[\boldsymbol{S} \subseteq \boldsymbol{T}] \\ &= 1 - 2\prod_{j=1}^{\ell-k}(1 - \Pr[j \in \boldsymbol{S} \wedge j \notin \boldsymbol{T}]) = 1 - 2\left(\frac{3}{4}\right)^{\ell-k} > \frac{2}{3} + \frac{1}{100} \text{ if } \ell - k \ge 7. \end{aligned}$$

Hence, the largeness property is satisfied.

Construction of $\boldsymbol{\alpha}$. Let $(\gamma, E) \in \Xi$, we define $\alpha(\gamma, E) = (\gamma_\alpha, E_\alpha)$, where E_α is a matrix with rows defined by vectors $(e'_1, \dots, e'_\ell) = (e_1, \dots, e_{\ell-k}, e_{\ell-k+1} + \sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j, e_{\ell-k+2}, \dots, e_\ell)$, and

$$\gamma_\alpha(i) = \begin{cases} \gamma(*) & \text{if } i = i_0 \\ \gamma(i_0) & \text{if } i = * \\ \gamma(i) & \text{otherwise} \end{cases}.$$

▷ Claim 32. α is a bijection from $\Xi \to \Xi$.

Proof. Notice that rows of E' form a basis since $\sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j \in \mathrm{Span}(e_1, \dots, e_{\ell-k})$. The mapping $\gamma \mapsto \gamma_\alpha$ is bijective since it just swaps $\gamma(i_0)$ and $\gamma(*)$. Since the condition on $\gamma(i_0)$ and $\gamma(*)$ does not change after application of α, we get that $\alpha(\Xi) \subseteq \Xi$. Notice that $\sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j = \sum_{j=1}^{\ell-k}(\gamma_\alpha(i_0)_j + \gamma_\alpha(*)_j)e'_j$, hence $\alpha(\gamma_\alpha, E_\alpha) = (\gamma, E)$, hence α is bijective. ◁

▷ Claim 33. For all $(\gamma, E) \in \Xi$ the following equalities hold

1. $R(\gamma_\alpha(i_0), 1, E_\alpha) = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\alpha(i_0), 0, E_\alpha) = R(\gamma(*), 0, E)$;
3. $R(\gamma_\alpha(*), 0, E_\alpha) = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\alpha(*), 1, E_\alpha) = R(\gamma(*), 1, E)$.

Proof. We use Claim 28. Let us denote $S := \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell) = \mathrm{Span}(e'_{\ell-k+2}, \dots, e'_\ell)$.

1. $R(\gamma_\alpha(i_0), 1, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(i_0)_j e'_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j + e_{\ell-k+1}\right) + S = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\alpha(i_0), 0, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(i_0)_j e'_j\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j\right) + S = R(\gamma(*), 0, E)$;
3. $R(\gamma_\alpha(*), 0, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(*)_j e'_j\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j\right) + S = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\alpha(*), 1, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(*)_j e'_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j + e_{\ell-k+1}\right) + S = R(\gamma(*), 1, E)$. ◁

Construction of β. For $(\gamma, E) \in \Xi$, we define $\beta(\gamma, E) = (\gamma_\beta, E_\beta)$, where $\gamma_\beta = \gamma_\alpha$ and E_β is defined below. Let $j_{\min} = \min\{j \in [\ell - k] : (\gamma(*))_j = 1 \wedge (\gamma(i_0))_j = 0\}$; $j_{\min}$ is correctly defined since $(\gamma, E) \in \Xi$. Now we define $E_\beta = (e''_1, \dots, e''_\ell)$:

$$e''_j = \begin{cases} e_j & \text{if } j \notin \{j_{\min}, \ell - k + 1\} \\ \sum_{i=1}^{\ell-k}(\gamma(*)_i + \gamma(i_0)_i)e_i & \text{if } j = \ell - k + 1 \\ e_{j_{\min}} + e_{\ell-k+1} & \text{if } j = j_{\min} \end{cases}.$$

▷ Claim 34. β is a bijection from $\Xi \to \Xi$.

Proof. Let us verify that β is *injective*. Given γ_β we can easily recover γ, hence we can recover $j_{\min}$ as well. Then

$$\sum_{i=1}^{\ell-k}(\gamma(i_0)_i + \gamma(*)_i)e''_i + e''_{\ell-k+1} = \sum_{i \in [\ell-k] \setminus \{j_{\min}\}} (\gamma(i_0)_i + \gamma(*)_i)e_i + \overbrace{e_{j_{\min}} + e_{\ell-k+1}}^{e''_{j_{\min}}} + e''_{\ell-k+1}$$

$$= e_{\ell-k+1} + \underbrace{\sum_{i \in [\ell-k] \setminus \{j_{\min}\}} (\gamma(i_0)_i + \gamma(*)_i)e_i + e_{j_{\min}}}_{e''_{\ell-k+1}} + e''_{\ell-k+1} = e_{\ell-k+1}.$$

Thus, we can uniquely recover $e_{\ell-k+1}$ and, hence, also recover $e_{j_{\min}} = e''_{j_{\min}} + e_{\ell-k+1}$; for $j \in [\ell] \setminus \{j_{\min}, \ell - k + 1\}$, $e_j = e''_j$. Hence, β is injective. Notice that since we represent $e_1, \dots, e_\ell$ as linear combinations of $e''_1, \dots, e''_\ell$, then $e''_1, \dots, e''_\ell$ is a basis, hence the matrix E_β is invertible. Thus, we verify that $\beta(\Xi) \subseteq \Xi$ and β is injective, hence β is bijective. ◁

▷ Claim 35. For all $(\gamma, E) \in \Xi$ the following equalities hold

1. $R(\gamma_\beta(i_0), 1, E_\beta) = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\beta(i_0), 0, E_\beta) = R(\gamma(*), 1, E)$;
3. $R(\gamma_\beta(*), 0, E_\beta) = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\beta(*), 1, E_\beta) = R(\gamma(*), 0, E)$;

Proof. We denote $S := \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell) = \mathrm{Span}(e''_{\ell-k+2}, \dots, e''_\ell)$. Recall that $\gamma(*)_{j_{\min}} = 1$ and $\gamma(i_0)_{j_{\min}} = 0$.

1. $R(\gamma_\beta(i_0), 1, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(i_0)_i e''_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + \sum_{i=1}^{\ell-k} (\gamma(*)_i + \gamma(i_0)_i) e_i + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + e_{\ell-k+1} + S = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\beta(i_0), 0, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(i_0)_i e''_i + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + S = R(\gamma(*), 1, E)$;
3. $R(\gamma_\beta(*), 0, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(*)_i e''_i + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + S = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\beta(*), 1, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(*)_i e''_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + S = R(\gamma(*), 0, E)$. ◁

Claims 33 and 35 imply the equations 2. ◀

5.4 Proof of Lemma 21

To prove Lemma 21 it is sufficient to prove the following:

▶ **Proposition 36.** *There exist matrices $T_1, \dots, T_k \in \mathbb{F}_2^{2^k \times k}$, such that*

- *for $\alpha_1, \dots, \alpha_k \in \{0, 1\}$ the matrix $\sum_{i=1}^k \alpha_i T_i$ is zero iff $\alpha_1 = \alpha_2 = \dots = \alpha_k = 0$, i.e. $T_1, \dots, T_k$ are linearly independent;*
- *For every non-zero matrix $M \in \mathrm{Span}(T_1, \dots, T_k)$, $M + K_k \in \mathbb{S}_k$.*

Proof of Lemma 21. Let for $i \in \{1, \dots, k-1\}$, $A_i(0) = T_i$ and $A_i(1)$ be the zero matrix. Let $A_k(0) = K_k + T_k$, $A_k(1) = K_k$. For each $b_1, \dots, b_k \in \{0, 1\}$, $\sum_{i=1}^k A_i(b_i) = \sum_{i=1}^k (1 - b_i) T_i + K_k$. Then $\sum_{i=1}^k A_i(1) = K_k$, and if for at least one $i \in [k]$, $b_i \neq 1$, then by the first condition of Proposition 36, $\sum_{i=1}^k (1 - b_i) T_i$ differs from zero, thus by the second condition of Proposition 36, $\sum_{i=1}^k A_i(b_i) \in \mathbb{S}_k$. ◀

Proof of Proposition 36. Let us prove the proposition by induction on k. We are going to prove a stronger statement: namely, we additionally require that for arbitrary non-zero matrix $M \in \mathrm{Span}(T_1, \dots, T_k)$ the set of even-indexed rows of $M + K_k \in \mathbb{S}_k$ coincide with the set of odd-indexed rows of this matrix with all bits flipped.

The base case $k = 1$. $T_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and $K_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. It is easy to verify that all conditions hold.

Induction step from k to $k+1$. Notice that $K_{k+1} = \begin{pmatrix} K_k & \mathbf{0}_{2^k \times 1} \\ K_k & \mathbf{1}_{2^k \times 1} \end{pmatrix}$. Let $T_1, \dots T_k$ be the matrices from induction hypothesis for k. Then define $T'_i = \begin{pmatrix} T_i & \mathbf{0}_{2^k \times 1} \\ T_i & \mathbf{0}_{2^k \times 1} \end{pmatrix}$ for $i \in [k]$ and $T'_{k+1} = \begin{pmatrix} \mathbf{0}_{2^k \times k} & z_0 \\ \mathbf{1}_{2^k \times k} & z_1 \end{pmatrix}$, where $z_0 = (0, 1, 0, 1, \dots, 0, 1)^T \in \{0, 1\}^{2^k \times 1}$, and $z_1 = (1, 0, 1, 0 \dots, 1, 0)^T \in \{0, 1\}^{2^k \times 1}$.

Let us verify that all conditions hold. First we show that the matrices $T'_1, T'_2, \dots, T'_{k+1}$ are linearly independent. Matrices $T'_1, T'_2, \dots, T'_k$ are linearly independent since they contain linearly independent blocks $T_1, \dots, T_k$. The matrix T'_{k+1} does not belong to $\mathrm{Span}(T'_1, \dots, T'_k)$, since the last column of T'_{k+1} is non-zero, but the last columns of all $T'_1, \dots, T'_k$ are zeros.

Let us check that for any non-zero matrix $M \in \mathrm{Span}(T'_1, \dots, T'_k, T'_{k+1})$, the condition $M + K_{k+1} \in \mathbb{S}_{k+1}$ holds and the set of even-indexed rows of $M + K_{k+1}$ coincide with the set of odd-indexed rows of this matrix with all bits flipped. Let us analyze the cases:

1. Let M be a non-zero matrix from $\mathrm{Span}(T'_1, \dots, T'_k)$. Then, M has form $\begin{pmatrix} M' & \mathbf{0}_{2^k\times 1} \\ M' & \mathbf{0}_{2^k\times 1}\end{pmatrix}$, where M' is a non-zero matrix from $\mathrm{Span}(T_1, \dots, T_k)$, thus $M' + K_k \in \mathbb{S}_k$. Then $M + K_{k+1} = \begin{pmatrix} M' + K_k & \mathbf{0}_{2^k\times 1} \\ M' + K_k & \mathbf{1}_{2^k\times 1}\end{pmatrix}$; it follows from the induction hypothesis that all rows of this matrix are distinct, i.e. $M + K_{k+1} \in \mathbb{S}_{k+1}$. In order to verify that the set of even-indexed rows of this matrix coincide with the set of odd-indexed rows with all bits flipped, observe that by induction hypothesis the first 2^{k-1} even-indexed rows of $M + K_{k+1}$ coincide with the last 2^{k-1} odd-indexed rows of $M + K_{k+1}$ with all bits flipped, and the first 2^{k-1} odd-indexed rows of $M + K_{k+1}$ coincide with the last 2^{k-1} even-indexed rows of $M + K_{k+1}$ with flipped bits.
2. $M = T'_{k+1}$, then $M + K_{k+1} = \begin{pmatrix} K_k & z_0 \\ \mathbf{1}_{2^k\times k} + K_k & z_0\end{pmatrix}$. Let us show that all rows of this matrix are distinct. The first 2^k rows start with 0 and are obtained by appending zeroes and ones to the rows of K_k in the alternating order. Since for every pair of coinciding rows of K_k they are adjacent, the first 2^k rows are distinct. The last 2^k rows start from one, so they differ from the first 2^k rows. The proof that they are distinct is the same as for the first 2^k rows. Observe that the $(2i-1)$th row of the matrix $M + K_{k+1}$ coincide with the $(2^k + 2i)$th row of $M + K_{k+1}$ with flipped bits, and the $(2i)$th row of $M + K_{k+1}$ coincide with the $(2^k + 2i - 1)$th row of $M + K_{k+1}$ with flipped bits for $i \in [2^k]$.
3. $M = R + T'_{k+1}$, where R is a non-zero matrix from $\mathrm{Span}(T'_1, \dots, T'_k)$. Let R have the form $\begin{pmatrix} R' & \mathbf{0}_{2^k\times 1} \\ R' & \mathbf{0}_{2^k\times 1}\end{pmatrix}$, where R' is a non-zero matrix from $\mathrm{Span}(T_1, \dots, T_k)$. Then $M + K_{k+1} = R + T'_{k+1} + K_{k+1} = \begin{pmatrix} R' + K_k & z_0 \\ \mathbf{1}_{2^k\times k} + R' + K_k & z_0\end{pmatrix}$. By the induction hypothesis, $R' + K_k \in \mathbb{S}_k$ and its even-indexed rows coincide with its odd-indexed rows with flipped bits. Then, all even-indexed rows of $M + K_{k+1}$ end with 0, the first 2^{k-1} of them are even-indexed rows of $R' + K_k$ with appended zero, and the last 2^{k-1} of them are even-indexed rows of $R' + K_k$ with all bits flipped and appended 0. Then, by the induction hypothesis, the set of the former rows does not intersect with the set of the latter rows, therefore they are all distinct. By the same argument, all the rows of $M + K_{k+1}$ that end with 1 are distinct. Thus, $M + K_{k+1} \in \mathbb{S}_{k+1}$.
   Let us verify that the set of even-indexed rows of this matrix coincide with the set of odd-indexed rows of this matrix with all bits flipped. Observe that if the ith row of $R' + K_k$ coincides with the jth row of $R' + K_k$ with flipped bits, then the ith row of $M + K_{k+1}$ coincides with its jth row with flipped bits, and the $(2^k + i)$th row of $M + K_{k+1}$ coincides with its $(2^k + j)$th row with all bits flipped. The required property follows from the induction hypothesis. ◀

5.5 Corollaries

▶ **Corollary 37.** *If $k + 7 \le \ell$, then the size of any semantic $\mathrm{Res}(\mathrm{PC}_{k-1})$ tree-like refutation of $\mathrm{BPHP}^{2^\ell+2^k}_{2^\ell}$ is at least $2^{\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)}$. For $k = 2$, the size of any tree-like semantic $\mathrm{Res}(\oplus)$ refutation of $\mathrm{BPHP}^{2^\ell+4}_{2^\ell}$ is at least $2^{\Omega(2^\ell)}$.*

Proof. Follows from Theorem 3 and Lemma 1. ◀

▶ **Corollary 38.** *Let $2 \le k \le \ell - 7$ and S be the minimal size of tree-like refutation of $\varphi = \mathrm{BPHP}_{2^\ell}^{2^\ell + 2^k} \circ \oplus_k$ in the semantic proof system $\mathrm{T}^{cc}(k,c)$. Then $\log S \log\log S \ge c \cdot \Omega\left(\frac{2^{\ell/2}}{k 2^{3k/2}}\right)$. For $k = 2$, $\log S \log\log S \ge c \cdot \Omega\left(2^\ell\right)$.*

Proof. By Lemma 9, $R_{pub}^{1/3}(\mathrm{Search}(\varphi)) = \mathcal{O}\left(\frac{\log S \log\log S}{c}\right)$. We also know that

$$R_{pub}^{1/3}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell + 2^k}^{2^\ell} \circ \oplus_k\right)\right) \ge R_{pub}^{1/3}\left(\oplus_k \mathrm{Search}\left(\mathrm{BPHP}_{2^\ell + 2^k}^{2^\ell}\right)\right).$$

Now the statement follows from Theorem 3. ◀

6 Bit pigeonhole principle

6.1 Reduction from BPHP $\circ \oplus_k$ to BPHP

Let $T \subseteq X_1 \times X_2 \times \cdots \times X_k \times Y$ and $S \subseteq Z_1 \times Z_2 \times \cdots \times Z_k \times W$ be two relations. We say that S is many-one reducible to T if there are $k+1$ mappings $f_1 : X_1 \to Z_1$, $f_2 : X_2 \to Z_2, \ldots, f_k : X_k \to Z_k$ and $g : W \to Y$ such that if $(f_1(x_1), \ldots, f_k(x_k), y) \in T$ then $(x_1, \ldots, x_k, g(y)) \in S$.

▶ **Lemma 39.** *If S is many-one reducible to T, then $R_{1/3}^{pub}(S) \le R_{1/3}^{pub}(T)$.*

Proof. The ith party computes $f(x_j)$ for all $j \in [k] \setminus \{i\}$ and then all parties run the optimal protocol for T. As soon as all the parties learn an answer y they compute $g(y)$ without communication. ◀

Recall that $\mathrm{BPHP}_{2^n}^M$ encodes that there exist M different strings $s_1, s_2, \ldots, s_M$ from $\{0,1\}^n$. Let k be a positive integer. Let us define the partition Π_k of the variables of $\mathrm{BPHP}_{2^n}^M$ into k parts. Let $n = \ell k + r$ where $0 \le r < k$. For each $i \in [M]$ the row s_i is partitioned into k parts $s = s_i^{(1)} s_i^{(2)} \cdots s_i^{(k)}$ such that $|s_i^{(t)}| = \ell + 1$ if $t \le r$, and $|s_i^{(t)}| = \ell$ if $t > r$. The partition Π_k of the variables of $\mathrm{BPHP}_{2^n}^M$ into k parts is the following: the tth part consists of the variables $s_1^{(t)}, s_2^{(t)}, \ldots, s_M^{(t)}$.

We consider a search problem $\mathrm{SearchPair}_{2^n}^M$: given the values of the variables of $\mathrm{BPHP}_{2^n}^M$, that are partitioned according to Π_k find a pair of distinct indices $i, j \in [M]$, such that the values of s_i and s_j coincide.

▶ **Proposition 40.** *The relation $\mathrm{SearchPair}_{2^n}^M$ is many-one reducible to $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^M\right)$ with variables partitioned according to Π_k.*

Proof. The proof is straightforward. ◀

▶ **Theorem 41.** *$\oplus_k \mathrm{BPHP}_{2^\ell}^m$ is many-one reducible to $\mathrm{SearchPair}_{2^{k\ell}}^{m \cdot 2^{(k-1)\ell}}$.*

Proof. Let us denote $M = m \cdot 2^{(k-1)\ell}$. Consider a set $Z = \left\{(y_1, y_2, \ldots, y_k) \in (\mathbb{F}_2^\ell)^k \mid \sum_i y_i = 0\right\}$. It is easy to see that $|Z| = 2^{(k-1)\ell}$. Let φ be a bijection between $[M]$ and $Z \times [m]$.

Let for $i \in [m]$ and $t \in [k]$, $x_i^{(t)}$ denote the ith string of the tth party in the communication problem $\oplus_k \mathrm{BPHP}_{2^\ell}^m$. Let $x_i := (x_i^{(1)}, \ldots, x_i^{(k)})$.

For every $t \in [k]$ we define f_t as follows: $f_t\left(x_1^{(t)}, \ldots, x_m^{(t)}\right)$ is a sequence of rows $r_1^{(t)}, r_2^{(t)}, \ldots, r_M^{(t)}$ such that for all $i \in [M]$, $r_i^{(t)} = z_t + x_j^{(t)}$, where $(z, j) = \varphi(i)$ for all $z \in Z$ and $j \in [m]$ (recall that $z \in Z$ is divided on k parts of equal lengths and z_t denotes the tth part).

Let us construct the function g from the definition of the reduction.

Let $q, w \in [M]$ and $q \ne w$. Assume that $\varphi(q) = (z, j_1)$ and $\varphi(w) = (z, j_2)$. We define $g(q, w) := (j_1, j_2)$.

Let us verify that $f_1, f_2, \dots, f_k$ and g define a reduction. Let $q, w \in M$ be a pair of different numbers such that the assignment $\alpha := \left\{ s_i \leftarrow r_i^{(1)} r_i^{(2)} \dots r_i^{(k)} \mid i \in [M] \right\}$ satisfies $s_q = s_w$. Assume that $g(q, w) = (j_1, j_2)$. We need to verify that $j_1 \neq j_2$ and $\sum_{t=1}^{k} x_{j_1}^{(t)} = \sum_{t=1}^{k} x_{j_2}^{(t)}$.

Notice that under the assignment α the value of s_q is $x_{j_1} + z$ and the value of s_w is $x_{j_2} + y$, where $j_1, j_2 \in [m]$ and $z, y \in Z$ such that $(z, j_1) = \varphi(q)$ and $(y, j_2) = \varphi(w)$. If $j_1 = j_2$, then $x_{j_1} + z = x_{j_2} + y$ implies $z = y$. Since φ is a bijection, we get $q = w$. Thus, $j_1 \neq j_2$.

For each $t \in [k]$, the following equality holds.

$$z_t + x_{j_1}^{(t)} = y_t + x_{j_2}^{(t)} \tag{3}$$

If we sum up equations (3) for all $t \in [k]$ and use that $y, z \in Z$, we get $\sum_{t=1}^{k} x_{j_1}^{(t)} = \sum_{t=1}^{k} x_{j_2}^{(t)}$. Hence, (j_1, j_2) is a correct answer for $\oplus_k \mathrm{BPHP}_{2^\ell}^{m}$. ◀

The following proposition deals with the case, where the number of bits is not divisible by k.

▶ **Proposition 42.** *Let $n = k\ell + r$, where $0 \le r < k$. Let $M > 2^{k\ell}$. Then* $\mathrm{SearchPair}_{2^{k\ell}}^{M}$ *is many-one reducible to* $\mathrm{SearchPair}_{2^n}^{M2^r}$.

Proof. Let $x_1, x_2, \dots, x_M$ be the input of $\mathrm{SearchPair}_{2^{k\ell}}^{M}$, let $x_j^{(t)}$ be the tth part of the row x_j according to the partition Π_k. Given this input we construct an input for $\mathrm{SearchPair}_{2^n}^{M2^r}$. Let τ be a bijection between $[M] \times \{0,1\}^r$ and $[M2^r]$.

For each $i \in [M]$ we construct 2^r rows $y_{\tau(i,\alpha)}$ one for each $\alpha \in \{0,1\}^r$. Let Π_k partition a row $y_{\tau(i,\alpha)}$ into the following parts: $y_{\tau(i,\alpha)}^{(1)} y_{\tau(i,\alpha)}^{(2)} \cdots y_{\tau(i,\alpha)}^{(k)}$. Let

$$y_{\tau(i,\alpha)}^{(t)} = \begin{cases} x_i^{(t)} & \text{if } t > r \\ x_i^{(t)} \alpha_t & \text{if } 0 \le t \le r \end{cases}.$$

Now we can define the function $f_t(x_1^{(t)}, \dots, x_M^{(t)})$ as $y_{\tau(i,\alpha)}^{(t)}$ for each $i \in M$ and $\alpha \in \{0,1\}^r$ and $t \in [k]$ Observe that for each $i \in [M]$ the rows $y_{i,\alpha}$ for $\alpha \in \{0,1\}^r$ are distinct. That allows us to define the function g as $g(\tau(i_1, \alpha_1), \tau(i_2, \alpha_2)) = (i_1, i_2)$. All the required properties can be easily verified. ◀

▶ **Theorem 4.** *Let $M = 2^n + 2^{k+n-\lfloor n/k \rfloor}$ and $n \ge k(k+7)$. If variables of* $\mathrm{BPHP}_{2^n}^{M}$ *are partitioned according Π_k, then $R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) = \Omega\left(\frac{2^{n/2k - 3k/2}}{k}\right)$.*

For $k = 2$ a stronger bound holds: $R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) = \Omega(2^{n/2})$.

Proof. Let $\ell = \lfloor n/k \rfloor$ and $r = n - \ell k$.

$$\begin{aligned}
R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) &= R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{(2^k+2^\ell)2^{(k-1)\ell+r}}\right)\right) \\
&\overset{\text{(Proposition 40)}}{\ge} R_{1/3}^{pub}\left(\mathrm{SearchPair}_{2^n}^{(2^k+2^\ell)2^{(k-1)\ell+r}}\right) \\
&\overset{\text{(Proposition 42)}}{\ge} R_{1/3}^{pub}\left(\mathrm{SearchPair}_{2^{k\ell}}^{(2^k+2^\ell)2^{(k-1)\ell}}\right) \\
&\overset{\text{(Theorem 41)}}{\ge} R_{1/3}^{pub}\left(\oplus_k \mathrm{BPHP}_{2^\ell}^{2^k+2^\ell}\right) \overset{\text{(Corollary 19)}}{=} \Omega\left(\frac{2^{\ell/2 - 3k/2}}{k}\right) = \Omega\left(\frac{2^{n/2k - 3k/2}}{k}\right).
\end{aligned}$$

The case of $k = 2$ can be treated in the same way, the only difference is in the application of Corollary 19. ◀

6.2 Upper bound for communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}^{m}_{2^n}\right)$

▶ **Proposition 5.** *For $M > 2^n$ and $k \in \{2, 3, \ldots, n\}$ there exists a* deterministic *NOF communication protocol for* $\mathrm{Search}\left(\mathrm{BPHP}^{M}_{2^n}\right)$ *w.r.t.* Π_k *transmitting* $\mathcal{O}\left(2^{\lceil n/k\rceil} \cdot \log M\right)$ *bits.*

Proof. The protocol is going to have only two active parties: the second party, which we call Alice, and the first party, which we call Bob. We are going to use that Alice can see the variables $s^{(1)}_1, \ldots, s^{(1)}_M$ and that Bob can see all other variables.

Let us denote $\bar{s}^{(1)}_i = s^{(2)}_i s^{(3)}_i \ldots s^{(k)}_i \in \{0,1\}^{n-\lceil n/k\rceil}$ the bits Bob sees in the ith line for $i \in [M]$. Bob finds a value $\alpha \in \{0,1\}^{n-\lceil n/k\rceil}$ such that the size of the set $S_\alpha = \left\{i \in [M] \mid \bar{s}^{(1)}_i = \alpha\right\}$ is larger than $2^{\lceil n/k\rceil}$. Such α exists since $M > 2^n$. Bob then picks an arbitrary subset S' of S_α of size $2^{\lceil n/k\rceil} + 1$ and sends the description of S' to Alice using $\left(2^{\lceil n/k\rceil} + 1\right) \cdot \lceil \log_2 M\rceil$ bits. Then, by the pigeonhole principle there exists $i \neq j \in S'$ such that $s^{(1)}_i = s^{(1)}_j$. Alice and Bob then spend $\mathcal{O}(\log M + n)$ bits transmitting indices i and j and all the values of the ith and jth lines to each other. Both of them then find the falsified clause of $\mathrm{BPHP}^{M}_{2^n}$ with no communication because it only depends on variables s_i and s_j and broadcast its description to all of the parties using an additional $\mathcal{O}(n + \log M)$ bits. ◀

For $k = 2$ this upper bound coincides with the lower bound given by Corollary 19 up to a logarithmic factor. For the larger value of k the upper bound and the lower bound are polynomially related. This upper bound shows that the dependence on k in the lower bound is not an artifact of the proof, but a genuine phenomenon.

6.3 Short $\mathrm{Th}(\log n)$ proof of BPHP^{m}_{n}

In this section we give a short tree-like $\mathrm{Th}(\log n)$ refutation of the bit pigeonhole principle BPHP^{m}_{n}. This observation is similar to the one of [5] that converts a resolution proof of the unary encoding of the pigeonhole principle PHP^{m}_{n} to a proof of BPHP^{m}_{n} in $\mathrm{Res}(\log n)$.

Namely we prove the following:

▶ **Proposition 43.** *If there exists a tree-like* $\mathrm{Th}(1)$*-refutation of* $\mathrm{PHP}^{m}_{2^\ell}$ *of size S. Then there exists a tree-like* $\mathrm{Th}(\ell)$*-refutation of* $\mathrm{BPHP}^{m}_{2^\ell}$ *of size $\mathcal{O}(S)$.*

Proof. Let $p_{i,j}$ for $i \in [m]$ and $j \in [2^\ell]$ be a variable of $\mathrm{PHP}^{m}_{2^\ell}$ indicating that the ith pigeon flies to the jth hole. Let $s_{i,k}$ for $i \in [m]$, $k \in [\ell]$ be a variable of $\mathrm{BPHP}^{m}_{2^\ell}$ indicating the ℓth bit of the ith string s_i.

Let $Q_j(x_1, x_2, \ldots, x_\ell)$ for $j \in [2^k]$ be a multilinear polynomial over reals such that for all $a_1, a_2, \ldots, a_\ell \in \{0,1\}^\ell$, $Q_j(a_1, a_2, \ldots, a_\ell) = 1$ if $(a_1, a_2, \ldots, a_\ell) = \mathtt{bin}_\ell(j-1)$ and $Q_j(a_1, a_2, \ldots, a_\ell) = 0$ otherwise. We ma define Q_j as follows $Q_j(x_1, \ldots, x_\ell) = \prod_{k=1}^{\ell}(1 - x_k + \alpha_\ell)$ for $i \in [m]$, $j \in [2^k]$, where $\alpha = \mathtt{bin}_\ell(j-1)$. By the construction $\deg(Q_j) = \ell$.

Let $P_{i,j} = Q_j(s_{i,1}, s_{i,2}, \ldots, s_{i,\ell})$.

Consider a tree-like $\mathrm{Th}(1)$-refutation of $\mathrm{PHP}^{m}_{2^\ell}$ of size S: $f_1 \geq 0, f_2 \geq 0, \ldots, f_S \geq 0$, where f_i are linear real polynomials over variables $p_{i,j}$ and $f_S \geq 0$ is unsatisfiable on Boolean cube. For each of the inequalities on the following conditions hold:

(a) $f_i \geq 0$ is semantically implied by $f_j \geq 0$ and $f_k \geq 0$ *on the Boolean cube* for $j, k < i$.
(b) f_i is a linear representation of an axiom of $\mathrm{PHP}^{m}_{2^\ell}$;

Let F_i be a polynomial obtained of substitution $p_{j,k} := P_{j,k}$ to f_i for all $j \in [m]$; $k \in [2^\ell]$. Consider a sequence of inequalities $F_1 \geq 0, \ldots, F_S \geq 0$. Observe that $F_S \geq 0$ is unsatisfiable on the Boolean cube since $P_{i,j} \in \{0,1\}$ on the Boolean cube. Let us verify that the sequence $F_1 \geq 0, \ldots, F_S \geq 0$ may be extended to a correct tree-like $\mathrm{Th}(\ell)$ refutation of $\mathrm{BPHP}^{m}_{2^\ell}$:

(a) If $f_i \geq 0$ is semantically implied by $f_j \geq 0$ and $f_k \geq 0$, then $F_i \geq 0$ is also implied by $F_j \geq 0$ and $F_k \geq 0$, since $P_{i,j}$ is Boolean on the Boolean cube.

(b) If f_i is a linear representation of a

hole axiom then $f_i \geq 0$ is equivalent to the function $(1 - p_{a,b}) + (1 - p_{c,b}) \geq 1$ on $\{0,1\}^{\mathrm{Vars}(\mathrm{PHP}^m_{2^\ell})}$ for $a, c \in [m]$, $b \in [2^\ell]$. Thus $F_i \geq 0$ is also equivalent to $(1 - P_{a,b}) + (1 - P_{c,b}) \geq 1$ on the Boolean cube. Observe that the restriction of $(1 - P_{a,b}) + (1 - P_{c,b}) \geq 1$ to the Boolean cube coincides with the predicate $s_a \neq \mathtt{bin}_\ell(b) \vee s_c \neq \mathtt{bin}_\ell(b)$ which is an axiom of $\mathrm{BPHP}^m_{2^\ell}$.

pigeon axiom then $f_i \geq 0$ is equivalent to $\sum_{j=1}^{2^\ell} p_{a,j} \geq 1$ on the Boolean cube for some $a \in [m]$. Thus $F_i \geq 0$ is equivalent to $\sum_{j=1}^{2^\ell} P_{a,j} \geq 1$ on $\{0,1\}^{\mathrm{Vars}(\mathrm{BPHP}^m_{2^\ell})}$. Observe that the latter inequality is identically true, since $P_{a,j}$ is equivalent to $s_a = \mathtt{bin}_\ell(j-1)$, so for exactly one value of $j \in [2^\ell]$, $P_{a,j} = 1$. Since $F_i \geq 0$ is identically true it can be semantically derived from two arbitrary axioms of $\mathrm{BPHP}^m_{2^\ell}$.

It is easy to see that the size of the resulting refutation is at most $3S$. ◀

▶ **Proposition 44** ([4]). *For $m > n$ there exists a tree-like Cutting Planes (which is a subsystem of* $\mathrm{Th}(1)$*) refutation of* PHP^m_n *of size* $\mathcal{O}(m^2 n)$ *.*

▶ **Proposition 6.** *For $m > 2^\ell$ there exists a tree-like* $\mathrm{Th}(\ell)$ *refutation of* $\mathrm{BPHP}^m_{2^\ell}$ *of size* $\mathcal{O}(m^2 \cdot 2^\ell)$.

Proof. Follows from Propositions 43 and 44. ◀

References

1. Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007.
2. Paul Beame and Søren Riis. More on the relative strength of counting principles. In Paul Beam and Samuel R. Buss, editors, *Proof Complexity and Feasible Arithmetics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, April 21-24, 1996*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 13–35. DIMACS/AMS, 1996. `doi:10.1090/dimacs/039/02`.
3. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.
4. William Cook, Collette R. Coullard, and Gy. Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.
5. Stefan S. Dantchev, Nicola Galesi, and Barnaby Martin. Resolution and the binary encoding of combinatorial principles. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 6:1–6:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.CCC.2019.6`.
6. Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random Θ(log n)-CNFs are hard for cutting planes. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 109–120. IEEE Computer Society, 2017. `doi:10.1109/FOCS.2017.19`.
7. Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911. ACM, 2018. `doi:10.1145/3188745.3188838`.

8 Michal Garlík and Leszek Aleksander Kolodziejczyk. Some subsystems of constant-depth frege with parity. *ACM Trans. Comput. Log.*, 19(4):29:1–29:34, 2018. doi:10.1145/3243126.

9 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 847–856, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591838.

10 Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

11 Pavel Hrubes and Pavel Pudlák. A note on monotone real circuits. *Inf. Process. Lett.*, 131:15–19, 2018. doi:10.1016/j.ipl.2017.11.002.

12 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 233–248. ACM, 2012. doi:10.1145/2213977.2214000.

13 Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 220–228. IEEE Computer Society, 1994. doi:10.1109/LICS.1994.316069.

14 Dmitry Itsykson and Dmitry Sokolov. Lower bounds for splittings by linear combinations. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 2014. doi:10.1007/978-3-662-44465-8_32.

15 Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1), January 2020. doi:10.1016/j.apal.2019.102722.

16 Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, 1992. doi:10.1137/0405044.

17 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.

18 H. Kesten. An introduction to probability theory and its applications, volume i, (william feller). *SIAM Review*, 11(1):96–96, 1969. doi:10.1137/1011021.

19 Erfan Khaniki. On proof complexity of resolution over polynomial calculus. *Electronic Colloquium on Computational Complexity (ECCC)*, 27:34, 2020. URL: https://eccc.weizmann.ac.il/report/2020/034.

20 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997. doi:10.2307/2275541.

21 Jan Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. *J. Symb. Log.*, 73(1):227–237, 2008. doi:10.2178/jsl/1208358751.

22 Jan Krajíček. Randomized feasible interpolation and monotone circuits with a local oracle. *J. Mathematical Logic*, 18(2):1850012:1–1850012:27, 2018. doi:10.1142/S0219061318500125.

23 Jan Krajíček. *Proof complexity*, volume 170. Cambridge University Press, 2019.

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

25 Edward I Nechiporuk. A boolean function. *Engl. transl. in Sov. Phys. Dokl.*, 10:591–593, 1966.

26 Vsevolod Oparin. Tight upper bound on splitting by linear combinations for pigeonhole principle. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 77–84. Springer, 2016. doi:10.1007/978-3-319-40970-2_6.

27 Fedor Part and Iddo Tzameret. Resolution with counting: Dag-like lower bounds and different moduli. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 19:1–19:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.19.

28 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

29 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

30 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684.

31 Alexander A. Sherstov. The multiparty communication complexity of set disjointness. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 525–548, 2012.

32 Alexander A. Sherstov. Communication lower bounds using directional derivatives. *J. ACM*, 61(6), December 2014. doi:10.1145/2629334.

33 Dmitry Sokolov. Dag-like communication and its applications. In *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, pages 294–307, 2017. doi:10.1007/978-3-319-58747-9_26.

A Proof of Lemma 12

▶ **Lemma 12.** *Let T be a binary tree with m vertices such that the ith vertex is labeled with $a_i \in \{0,1\}$ with the* hereditary property*: for each inner vertex i with direct descendants c_1 and c_2, if $a_i = 1$, then $a_{c_1} = 1$ or $a_{c_2} = 1$. We also assume that if r is the root of T, then $a_r = 1$. Assume that we have a one-sided bounded error oracle access to a_i i.e. if we request a value of a_i and $a_i = 0$ we get 1 with probability at most $\frac{1}{2}$ and 0 with probability at least $\frac{1}{2}$; if $a_i = 1$ we get 1 with probability 1. Then there exists an algorithm $\mathcal{A}$ that with probability at least $\frac{2}{3}$ returns a leaf ℓ of T with $a_\ell = 1$ and makes $\mathcal{O}(\log m)$ oracle queries to $a_1, \ldots, a_m$.*

Proof of Lemma 12. For a tree F we denote by $|F|$ the number of nodes in F and for a node v of F we denote by $\textsc{Subtree}(F, v)$ the subtree of F with root v. Let $\texttt{Oracle}(i)$ be the oracle function returning the correct value of a_i with probability at least $\frac{9}{10}$. We can implement such a function using the majority vote of a constant number of initial oracle queries. Let C be a constant; an appropriate value of C we choose later. Consider Algorithm 2 on the following page.

We claim that at any iteration T_i has the hereditary property. This is the case in the beginning and if i decreases at some iteration, then the next T_i was considered at an earlier iteration. Otherwise, the next T_i is either a subtree of the current T_i (in that case the hereditary property is clearly maintained), or is obtained by removal a subtree with 0-labeled root (here we use that the oracle has a one-sided error) from the previous T_i (the hereditary property is also maintained in that case).

We first consider a variant of the algorithm that works infinitely long (i.e., $C = +\infty$) and compute the expected number of the first iteration such that T_i consists of a single 1-labeled leaf of T. Notice that after the first such iteration the value of T_i stays the same for all further iterations. We show that that the expected value is at most $C \log m$ for some constant C. Then by running the algorithm for $3C\lceil \log m \rceil$ iterations we obtain the required error probability by Markov's inequality.

Algorithm 2 Search for 1-leaf.

$T_0 := T$ ▷ Initialize the tree
$i := 0$
for $j := 1$ to $3C\lceil \log_{3/2} m \rceil$ **do**
  $r :=$ root of T_i
  if `Oracle`$(r) = 0$ **then**
    $i := \max\{0, i-1\}$ ▷ Backtrack since the current tree may not contain a 1-leaf
  else if $|T_i| \neq 1$ **then**
    $v :=$ a centroid node of T_i ▷ i.e. such that $|\textsc{Subtree}(T_i, v)| \in \left[\frac{1}{3}|T_i|, \frac{2}{3}|T_i|\right]$
    if `Oracle`$(v) = 1$ **then**
      $T_{i+1} := \textsc{Subtree}(T_i, v)$
    else
      $T_{i+1} := T_i - \textsc{Subtree}(T_i, v)$ ▷ T_{i+1} is obtained from T_i by the deletion of $\textsc{Subtree}(T_i, v)$
    $i := i + 1$
return the only node of T_i, if $|T_i| = 1$

Let $\mathtt{T}(j)$ denote the value of T_i before the start of jth iteration, $i(j)$ denote i at the start of jth iteration and $r(j)$ denote the root of $\mathtt{T}(j)$. Notice that if $a_{r(j)} = 1$, then for every $j' > j$, $\mathtt{T}(j')$ is a subtree of $\mathtt{T}(j)$, since the algorithm never backtracks if the true value of the roots label is 1. Hence, if $a_{r(j)} = a_{r(j')} = 1$ for some $j < j'$, then $i(j) \leq i(j')$.

Let us consider a sequence $j_1, j_2, j_3 \ldots$, where $j_1 = 0$, $j_s = \min\{j \mid a_{r(j)} = 1 \wedge j > j_{s-1} \wedge i(j) > i(j_{s-1})\}$, if such minimum exists.

Let us consider the iterations from j_s till $j_{s+1} - 1$. We consider the random variables $Y_{j_s}, Y_{j_s+1}, \ldots Y_{j_{s+1}-1}$ corresponding to these iterations with the following properties:

- If $\mathtt{T}(j)$ coincides with $\mathtt{T}(j_s)$, then its root is labeled with 1. Then $Y_j = -1$ if the second oracle query returns the correct answer and $Y_j = 1$ if the answer it incorrect. Notice that $\Pr[Y_j = -1] \geq \frac{9}{10}$.
- If the root of $\mathtt{T}(j)$ is labeled with zero, then $Y_j = -1$, if the first oracle query returns the correct answer (i.e. the algorithm backtracks). Otherwise, if $\mathtt{T}(j)$ consists of a single node $Y_j = 0$. Otherwise, if the root of $\mathtt{T}(j+1)$ is labeled with 0, then $Y_j = 1$. If it is labeled with 1, then $Y_j = -\infty$. Notice that $\Pr[Y_j \leq -1] \geq \frac{9}{10}$.

Notice that, $j_{s+1} = j_s + \min\{k \mid \sum_{j=j_s}^{j_s+k-1} Y_j \leq -1\}$. In order to estimate the expected value of $j_{s+1} - j_s$ we consider an auxiliary random variables $X_{j_s}, X_{j_s+1}, \ldots, X_{j_{s+1}-1}$, defined as $X_j = \begin{cases} 1, & \text{if } Y_j \geq 0 \\ -1, & \text{if } Y_j < 0 \end{cases}$. Notice then $\sum_{j=j_s}^{j_s+k-1} Y_j \leq \sum_{j=j_s}^{j_s+k-1} X_j$. We can apply the following fact about random walks in a straight line to the random variables X_j:

▶ **Theorem 45** (Section XII.2 of [18]). *Let $X_1, X_2, \ldots$ be a sequence of independent random variables that take value in $\{-1, 1\}$. Assume that for all i, $\Pr[X_i = 1] \leq \frac{1}{10}$ and $\Pr[X_i = -1] \geq \frac{9}{10}$. Let M be a random variable that equals the minimal natural number k such that $\sum_{i=1}^{k} X_i = -1$. Then the expected value of M is at most C, where $C \in \mathbb{R}$ is an absolute constant.*

Fact 45 implies that $\mathbb{E}[j_{s+1} - j_s] \leq C$. Then $\mathbb{E}[j_s] = \mathbb{E}[j_s - j_{s-1} + (j_{s-1} - j_{s-2}) + \cdots + (j_2 - j_1) + (j_1 - j_0)] \leq sC$. Thus, by Markov's inequality $\Pr[j_s \leq 3sC] \geq \frac{2}{3}$. Since $|T_{j_s}| \leq \left(\frac{2}{3}\right)^s |T_{j_0}|$, the algorithm that runs for $3C\lceil \log_{3/2} m \rceil$ iterations terminates in a 1-labeled leaf with probability at least $\frac{2}{3}$. ◀