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 (\(\varphi\)) is defined for every unsatisfiable CNF \(\varphi\): an assignment to the variables of \(\varphi\) is partitioned among the communicating parties, they are to find a clause of \(\varphi\) falsified by this assignment. Lower bounds on the randomized \(k\)-party communication complexity of Search (\(\varphi\)) in the number-on-forehead (NOF) model imply tree-size lower bounds, rank lower bounds, and size-space tradeoffs for the formula \(\varphi\) in the semantic proof system \(T^{cc}(k,c)\) 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 (\(\varphi\)) (e.g. [1, 9, 13]) are realized on ad-hoc formulas (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 \(Res(\oplus)\) that operates with disjunctions of linear equalities over \(F_2\) [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 \(Res(\oplus)\)-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 \(Res(\oplus)\)-refutations of \(PM_{K^n+2n}\).

Then we apply our approach for \(k\)-party communication complexity in the NOF model and obtain a \(\Omega(\frac{1}{k^2}n/2n)\) lower bound on the randomized \(k\)-party communication complexity of Search (BPHP\(_M^{2^n}\)) w.r.t. to some natural partition of the variables, where 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 Th(k) proofs, where Th(k) is the semantic proof system operating with polynomial inequalities of degree at most \(k\) and \(k = O((log^1 n)\) for some \(\epsilon > 0\). We also show that BPHP\(_M^{2^n+1}\) superpolynomially separates tree-like Th(log\(^1 m\)) from tree-like 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


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.

© Dmitry Itsykson and Artur Riazanov; licensed under Creative Commons License CC-BY 4.0

96th 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 \( \varphi \) defines a search problem \( \text{Search}(\varphi) \): the values of the variables of \( \varphi \) are partitioned between the parties of the protocol in some way, the participants are to find a clause of \( \varphi \) 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 \( \varphi \) of size \( S \) can be translated to a dag-like communication protocol for \( \text{Search}(\varphi) \) 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 [29], 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 \( \text{Search}(\varphi) \) in the model from the left column of Table 1 implies a lower bound on the size of tree-like refutations of \( \varphi \) in the corresponding proof system from the right column as well as a lower bound on the size of dag-like refutation of \( \varphi \) 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 \( \varphi \) of size \( S \) (or by a realization of a dag-like refutation of \( \varphi \) in size \( S \) within small space), one constructs a communication protocol for \( \text{Search}(\varphi) \) with
Table 1 Correspondence between communication models and proof systems.

<table>
<thead>
<tr>
<th>Communication model</th>
<th>Proof systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomized two-party protocols</td>
<td>Res(⊕) [15]. Proof lines in Res(⊕) are disjunctions of linear equations over $\mathbb{F}_2$.</td>
</tr>
<tr>
<td>Real $k$-party protocols in the number-on-head (NOF) model</td>
<td>Semantic $\text{Th}(k-1)$ [1]. Proof lines in $\text{Th}(k-1)$ 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.</td>
</tr>
<tr>
<td>Randomized $k$-party protocols in the NOF model</td>
<td>Semantic $\text{Tcc}(k,c)$. Proof lines in $\text{Tcc}(k,c)$ are arbitrary predicates that can be computed with $k$-party randomized communication cost at most $c$ in the NOF model. $\text{Tcc}(k,c)$ for small $c$ simulates $\text{Th}(k-1)$ and Res($\text{PC}_{k-1}$). Proof lines in Res($\text{PC}_d$) [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($\text{PC}_1$) coincides with Res($\oplus$).</td>
</tr>
</tbody>
</table>

The communication complexity $O(\log S \log \log S \cdot c)^1$ for an arbitrary partition of the variables of $\varphi$ 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 $(\varphi)$ for some fixed partition of variables between the parties.

Proving lower bounds for the communication complexity of Search $(\varphi)$ is not trivial since a lower bound on Search $(\varphi)$ 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 $(\varphi)$ are shown for artificial formulas $\varphi$ that are constructed as follows: take a standard formula $\psi$ 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 $(T(G) \circ \land_k)$, where $T(G)$ is an unsatisfiable Tseitin formula based on a special expander $G$ and $\land_k$ is the conjunction of $k$ variables, and the $i$th party has the $i$th argument of each instance of $\land_k$ 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 $(\varphi)$ 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 $O(\log S \cdot c)$
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 $\text{Res}(\oplus)$-refutations of the perfect matching principle, while the known lower bound techniques for tree-like $\text{Res}(\oplus)$ 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 $\text{Search} \left( \text{BPHP}_{2^m+2^{\Omega(1/k)}} \right)$, where $\text{BPHP}_{2^m}$ 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 $j$th part of every string is written on the forehead of the $j$th party. In particular, the latter result implies that the bit pigeonhole principle is hard for tree-like $\text{Th}(k)$, so it is the first natural hard instance.

1.2 Search problem $\oplus_k \text{Search} (\varphi)$

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

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

In Section 3 we observe the following lemma.

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

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

One of the most important open questions in proof complexity is obtaining a superpolynomial lower bound for bounded-depth Frege with parity gates. $\text{Res}(\oplus)$ 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 $\text{Res}(\oplus)$ were proved by Itsykson and Sokolov [14, 15]. Itsykson and Sokolov have shown a lower bound $2^{\Omega(n)}$ on size of tree-like $\text{Res}(\oplus)$ refutations of Pigeonhole Principle ($\text{PHP}_n^m$) for arbitrary $m > n$ using generalized Prover-Delayer games. Oparin in [26] has shown a tight upper bound $2^{O(n)}$ for such refutations. A lower bound $2^{\Omega(n)}$ for functional pigeonhole principle ($\text{FPHP}_n^m$) for $m = O(n)$ can be shown using a connection between the size of tree-like $\text{Res}(\oplus)$ refutations and the degree of polynomial calculus refutations (over $\mathbb{F}_2$), observed by Garlik and Kołodziejczyk (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 $\text{Res}(\oplus)$ 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$_2$ has a polynomial-size tree-like resolution 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 resolution refutation of PM$_{K_{n+2,n}}$ is $2^{Ω(n)}$.

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

The formula PM$_{K_{n+2,n}}$ (however, in a different encoding) has a constant-degree derivation in Nullstellensatz over $\mathbb{F}_2$ [2]. 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 $n$ mod 4 and on the other hand it is $(n + 2)$ mod 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$_n^m$.

Since PM$_{K_{n+2,n}}$ has a tree-like Cutting Planes refutation of polynomial size and with polynomial coefficients, the problem $\text{Search}(\text{PM}_{K_{n+2,n}})$ has communication complexity $O(\log n)$ for any partition and thus can not yield a superpolynomial lower bound on size of tree-like resolution refutations. Therefore the methods previously used to establish tree-like resolution lower bounds fail for 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 $Ω(n)$ on the two-party bounded-error randomized communication complexity of $\oplus_2 \text{Search}(\text{PM}_{K_{n+2,n}})$. 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 $Ω(n)$ [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 $\oplus_2 \text{Search}(\text{PM}_{K_{n+2,n}})$. 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 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
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 $\text{DISJ}_n(x,y) = 0$. After that Alice and Bob run the communication protocol for $\oplus_2\text{Search}(\text{PM}_{K_4,+3,+1})$. 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 $\text{BPHP}_{2^\ell}^m$ 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 $\ell$ and $k$ be natural numbers such that $2 \leq k \leq \ell - 7$. Then the randomized communication complexity of $\oplus_k\text{Search}(\text{BPHP}_{2^\ell}^m+2^\ell)$ in the $k$-party NOF model is $\Omega \left( \frac{2^{\ell/2}}{k^{2^\ell/2}} \right)$. 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 $\text{Distinct}_{k,\ell}$ that is similar to the search problem $\oplus_k\text{Search}(\text{BPHP}_{2^\ell}^m)$. 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 $\text{UDISJ}_{k,n}$ is the promise version of the $k$-party set disjointness: the $i$th 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 $x_j^{(i)} = 1$ for all $i \in [k]$ under the promise that there is at most one such index $j$. We describe a randomized reduction from $\text{UDISJ}_{k,2^\ell-3+k+1}$ to $\oplus_k\text{Search}(\text{BPHP}_{2^\ell}^m)$ and then use the known lower bound on the communication complexity
of the former problem \cite{32}. First, we reduce UDISJ_{k,2^ℓ−k} to the problem Distinct_{k,ℓ}: the
ith of the parties of the UDISJ protocol generates a matrix \( D_i \) of size \( 2^k \times \ell \) such that the
matrix \( \sum_{i=1}^k D_i \) contains a pair of equal rows iff UDISJ_{k,2^ℓ−k} evaluates to 0. Moreover, the
matrix \( \sum_{i=1}^k D_i \) has additional properties:
= each of the \( 2^ℓ−k \) 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 \( j \in [2^ℓ−k] \), 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 \( \oplus_k \text{Search} \left( \text{BPHP}_{2^ℓ}^{2^ℓ+2^k} \right) \). 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 \( \oplus_k \text{Search} \left( \text{BPHP}_{2^ℓ}^{2^ℓ+2^k} \right) \) finds a pair of equal rows coming from a genuine block.

Theorem 3 and Lemma 1 immediately imply the lower bound \( \exp \left( \Omega \left( \frac{2^{2k/3}}{3\pi n^{2/3}} \right) \right) \) on the
size of tree-like Res (PC_{k−1}) refutations of BPHP_{2^ℓ}^{2^ℓ+2^k} (for \( k = 2 \) the stronger lower bound \( \Omega(2^k) \) holds).

1.4.2 Bit pigeonhole without \( \oplus \)
gadget
In Section 6 we present a pretty simple and nice reduction from \( \oplus_k \text{Search} \left( \text{BPHP}_{2^m}^{m} \right) \) to
Search \( \left( \text{BPHP}_{2^m}^{2^m,2^m,1+k} \right) \). Here we describe this reduction for \( k = 2 \). For a larger \( k \) the
proof is essentially the same. Let us reduce \( \oplus_2 \text{Search} \left( \text{BPHP}_{2^m}^{m} \right) \) to Search \( \left( \text{BPHP}_{2^m}^{2^m,2^m} \right) \). We denote the input of Alice in \( \oplus_2 \text{Search} \left( \text{BPHP}_{2^m}^{m} \right) \) as \( a_1, \ldots, a_m \in \mathbb{F}_2^m \) and the input of
Bob as \( b_1, \ldots, b_m \in \mathbb{F}_2^m \). Their goal is to find a clause of BPHP_{2^m} falsified by the assignment
\( a_1 + b_1, \ldots, a_m + b_m \). 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 \( O(n) \) bits. For each \( i \in [m] \), Alice and Bob generate \( 2^m \) strings from \( \mathbb{F}_2^m \): Alice generates \( a_i + z \) for each \( z \in \mathbb{F}_2^m \) and Bob generates \( b_i + z \) for each \( z \in \mathbb{F}_2^m \). 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 Search \( \left( \text{BPHP}_{2^m,2^m}^{2^m} \right) \) on an input where each line has the
form \( (a_i + z, b_i + z) \) for each \( i \in [m] \) and \( z \in \mathbb{F}_2^m \). Given a falsified clause of BPHP_{2^m,2^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 \( (i, z) \in [m] \times \mathbb{F}_2^m \) is used by Alice and
Bob exactly once.

Together with Theorem 3 this yields the following theorem.

\textbf{Theorem 4.} For \( n \geq k(k+7) \) the randomized \( k \)-party communication complexity of
Search \( \left( \text{BPHP}_{2^m}^{2^m,2^m,1+k} \right) \) is \( \Omega \left( \frac{2^{n/2k−3k/2}}{\ell} \right) \), where every string of BPHP is partitioned
into \( k \) almost equal contiguous parts such that each \( k \)th party has the \( j \)th 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 \( \exp \left( 2^{\Theta(n/k)} \right) \) on the size of tree-like
T^{\text{cc}}(k, c) (and Th(k − 1)) refutations of BPHP_{2^m}^{2^m,2^m+k−1}.
Hrubes and Pudlák [10] proved a lower bound on the complexity of dag-like two-party real communication protocols for Search\(\text{BPHP}\_{m}^{2\ell}\) 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\(\text{BPHP}\_{2n}^{2\ell}\). 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.

\begin{proposition}
For \(M > 2^n\) and \(k \in \{2, 3, \ldots, n\}\) there exists a deterministic NOF communication protocol for Search\(\text{BPHP}_{M}^{2n}\) with variables partitioned as in Theorem 4 transmitting \(O\left(2^{\lceil n/k \rceil} \cdot \log M\right)\) bits.
\end{proposition}

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

\begin{proposition}
For \(m > 2^\ell\) there exists a tree-like Th\((\ell)\) refutation of BPHP\(_{2\ell}^{2\ell}\) of size \(O(m^2 \cdot 2^\ell)\).
\end{proposition}

Proposition 6 and the result of Hrubes and Pudlák [10] imply that tree-like Th\((\log n)\) refutation:\(\text{BPHP}_{2\ell}^{2\ell}\) can not be simulated by dag-like cutting planes due to the lower bound by [10] as well as separations between tree-like Th\((k)\) for different values of \(k\).

1.5 Open questions

1. Is it possible to prove lower bounds on the randomized communication complexity of \(\oplus_2\text{Search}(\text{PM}_G)\) 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\((\varphi)\) problem, where \(n\) is the number of variables.

2. Is it true that our results extend to Res\((\text{PC}_d)\) over arbitrary finite fields?

3. Is our lower bound for tree-like Th\((k)\) refutation of BPHP\(_{2n}^m\) tight? Such upper bound would imply a superpolynomial separation between tree-like Th\((k)\) and dag-like cutting planes due to the lower bound by [10] as well as separations between tree-like 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 BPHP\(_{2n}^M\), for example with \(M = 2^{n+1}\)?

---

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

Notations

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

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 $y \in Y$, $R_{\text{pub}}^x(Q, x, y)$ 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 $R_{\text{pub}}^y(Q) := \max_{x \in X, y \in Y} R_{\text{pub}}^x(Q, x, y)$.

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

Let $f$ be a function from $X_1 \times X_2 \times \ldots \times X_k \rightarrow Y$. Then $R_{\text{pub}}^y(f)$ denotes $R_{\text{pub}}^y(Q_f)$, where $Q_f = \{(x_1, x_2, \ldots, x_k, y) \mid f(x_1, \ldots, x_k) = y\}$.

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

Let us define the communication promise problem $\text{UDISJ}_{k,n}$ 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 $\text{DISJ}_{k,n}(x_1, \ldots, x_k)$.
Proof Complexity of Natural Formulas via Communication Arguments

- **Theorem 7** ([31, 32]). $R_{pub}^{1/3}(\text{DISJ}_{k,n}) = \Omega\left(\frac{n^k}{2^n}\right)$.

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

- **Theorem 8** ([16]). $R_{pub}^{1/3}(\text{DISJ}_{n}) \geq R_{pub}^{1/3}(\text{DISJ}_{2,n}) = \Omega(n)$.

### Proof complexity

We consider refutational proof systems for the language of unsatisfiable CNF-formulas UNSAT. A refutation of $\varphi \in \text{UNSAT}$ in a proof system $\Pi$ is a sequence of Boolean functions (proof lines) such that each proof line either represents a clause of $\varphi$ 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 $\Pi$ 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 $L_1, \ldots, L_k$.

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 $A \lor B$ from the clauses $A \lor x$ and $A \lor \neg x$.

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 $\text{Res}(\oplus)$ 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 $\bigvee_{i \in P} x_i \lor \bigvee_{j \in N} \neg x_j$ can be represented by the linear clause $\bigvee_{i \in P}(x_i = 1) \lor \bigvee_{j \in N}(x_j = 0)$. For definition of syntactic version of $\text{Res}(\oplus)$ we refer to [15]; it is also proved there that syntactic and semantic $\text{Res}(\oplus)$ are polynomially equivalent.

We define semantic $\text{Res}(\text{PC}_d)$ 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 $\text{Res}(\text{PC}_1)$ is exactly semantic $\text{Res}(\oplus)$. For the definition of the syntactic version of $\text{Res}(\text{PC}_d)$ we refer to [19].

Following [1] we define $\text{Th}(k)$ 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 $\bigvee_{i \in P} x_i \lor \bigvee_{j \in N} \neg x_j$ 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 $\varphi$ we define the communication problem $\text{Search}(\varphi)$. $\text{Search}(\varphi)$ is the following problem: given an assignment of the variables of the unsatisfiable CNF $\varphi$, find a clause that is falsified by this assignment. It is assumed that variables of $\varphi$ are somehow partitioned between the parties.

Following the paper [9] we consider a semantic proof system $T^{cc}(k, c)$ that models many interesting syntactic and semantic proof systems. The proof lines in $T^{cc}(k, c)$ 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 $T^{cc}_{os}(k, c)$ that is a subsystem of $T^{cc}(k, c)$ 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, $T^{cc}(2, 2)$ simulates Resolution; $T^{cc}(2, O(1))$ simulates Res$(\oplus)$ [22, 15]; $T^{cc}(k, O(k^3 \log^2 n))$, where $n$ is the number of variables in a refuted formula, simulates Th$(k - 1)$ [9]. In Section 3 we show that $T^{cc}_{os}(d + 1, O(1))$ simulates Res$(PC_d)$.

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

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

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

Basic formulas

A CNF formula $\text{PHP}_n^m$ encodes the pigeonhole principle; $\text{PHP}_n^m$ 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. $\text{PHP}_n^m$ 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. $\text{PHP}_n^m$ is the conjunction of $\frac{m(m-1)}{2}$ hole axioms and $m$ pigeons axioms. For every $i \in [m]$ $\text{PHP}_n^m$ contains a pigeon axiom $[p_{i,1} \lor p_{i,2} \lor \cdots \lor p_{i,n}]$. And for every $j \in [n]$ and every $k \neq \ell \in [n]$, $\text{PHP}_n^m$ contains a hole axiom $[\neg p_{k,j} \lor \neg p_{\ell,j}]$. $\text{PHP}_n^m$ is unsatisfiable iff $m > n$.

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

$$\text{PM}_G = \bigwedge_{v \in V} \left( \bigvee_{e \text{ is incident to } v} x_v \right) \wedge \bigwedge_{v \in V} \left( \neg x_v \lor \neg x_{e_1} \lor x_{e_2} \right).$$

$\text{PM}_G$ 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 $\text{PM}_G$ has a tree-like $\text{Res}(\oplus)$ refutation of size $2^{O(n)}$.

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

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

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

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

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

### Lemma 1

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

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 $i$th 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 $A$ that with probability at least $\frac{2}{3}$ returns a leaf $\ell$ of $T$ with $a_\ell = 1$ and makes $O(\log m)$ oracle queries to $a_1, \ldots, a_m$.

### Proof

See Appendix A.

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

Let $\alpha_1, \ldots, \alpha_{d+1}$ be the assignments written on the foreheads of the $d+1$ parties. Let $\alpha = \sum_{i=1}^{d+1} \alpha_i$. Let $a_i = 1$ iff $\alpha$ 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 $j$th party has $a_j$ written on their forehead such that

- the protocol transmits $O(d)$ 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 $\oplus_{d+1} \text{Search}(\varphi)$ with communication cost $O(d \log m)$.

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 $O(d)$ bits of communication. Let $F_\ell = \bigvee_{j=1}^m (f_j = 1)$, where $f_1, \ldots, f_m$ are polynomials over $\mathbb{F}_2$ of degree at most $d$. Let $z_1, \ldots, z_n$ be the variables of $\varphi$. Let us introduce new variables $y_{1,1}, \ldots, y_{1,n}, \ldots, y_{d+1,1}, \ldots, y_{d+1,n}$ and assume that for
each \(i \in [d+1]\) the \(i\)th party has the value of variables \(y_{i,1}, y_{i,2}, \ldots, y_{i,n}\) written on the forehead or in other words \(\alpha_i\) assigns values of \(y_{i,1}, y_{i,2}, \ldots, y_{i,n}\). Let \(\vec{f}_j\) 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 \(j \in [t]\), \(\deg f_j = \deg \vec{f}_j \leq d\), we can represent \(\vec{f}_j = \vec{f}_j^{(1)} + \ldots + \vec{f}_j^{(d+1)}\) such that \(\vec{f}_j^{(s)}\) does not contain variables \(y_{s,1}, \ldots, y_{s,n}\) for each \(s \in [d+1]\). Then the \(i\)th party can compute \(\vec{f}_j^{(i)}(\alpha_1, \ldots, \alpha_{d+1}), \ldots, \vec{f}_j^{(i)}(\alpha_1, \ldots, \alpha_{d+1})\). Notice that \(\vec{F}_i = -\left(\bigwedge_{j=1}^t \left(f_j = 0\right)\right)\).

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 \(\sum_{j=1}^t e_j f_j(\alpha) = \sum_{j=1}^{d+1} \sum_{j=1}^t e_j \vec{f}_j^{(i)}\) with \(O(d)\) 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 \(F_t(\alpha) = 1\) then \(\Pr \left[ \sum_{j=1}^t e_j f_j(\alpha) \neq 0 \right] = \frac{1}{2}\) by the random subsum principle. If \(F_t(\alpha) = 0\), 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 \(\varphi\) has a tree-like \(T^\varphi_n(k, c)\) refutation of length \(\ell\), then for any \(k\)-partition of the variables, there is a randomized bounded-error \(k\)-party NOF protocol for \(\text{Search}(\varphi)\) with communication cost \(O(c \log \ell)\). 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 \(\text{Res}(\oplus)\) refutation of the formula \(\text{PM}_{K_{n+2,n}}\) is \(2^{\Omega(n)}\).

By Lemma 1, to prove Theorem 2 it is sufficient to show that \(R_{\text{pub}}^{3/2}(\oplus \text{Search}(\text{PM}_{K_{n+2,n}})) = \Omega(n)\).

Consider the communication problem \(\oplus \text{PM}^m_n\) 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.** \(R_{\text{pub}}^{3/2}(\oplus \text{Search}(\text{PM}_{K_{n+2,n}})) \geq R_{\text{pub}}^{3/2}(\oplus \text{PM}^n_{n+2})\).

**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.** \( R_{pub}^{1/3}(\oplus \text{PM}_{n+2}^n) = \Omega(n) \).

**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 \( \oplus \text{PM}_{n+1}^n \) can be used for \( \oplus \text{PM}_{n+2}^n \) 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 \( \oplus \text{PM}_{n+2}^n \) 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 \((r_1,c_1,r_2,c_2) \in [n+2] \times [n]\) such that \((X+Y)_{r_1,c_1} = (X+Y)_{r_2,c_2} = 1 \) and either \( r_1 = r_2 \) or \( r_1 \neq r_2 \) and \( c_1 \neq c_2 \). With \( O(1) \) bits of communication Alice and Bob can verify whether the answer of \( \mathcal{P}_0 \) is correct and return \( \bot \) (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 \( \oplus \text{PM}_{n+2}^n \) under the promise that \( X + Y \) does not contain all-zero rows and columns that uses \( O(k) \) bits of communication and returns a correct answer with probability at least \( 99/100 \) and \( \bot \) otherwise.

We are going to construct a protocol for \( \text{DISJ}_m \) transmitting \( O(k) \) bits, where \( m = n - 1 \). Since by Theorem 8 any protocol for \( \text{DISJ}_m \) transmits \( \Omega(m) \) bits, we conclude that \( k = \Omega(m) \). Let Alice’s input for \( \text{DISJ}_m \) be \( a_1, \ldots, a_m \) and Bob’s input be \( b_1, \ldots, b_m \).

**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

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

**Proof.** We simply present matrices that satisfy the conditions:

\[
A(0) = \begin{pmatrix}
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}; \quad A(1) = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix};
\]

\[
B(0) = \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}; \quad B(1) = \begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}.
\]

Notice that Lemma 17 immediately allows to reduce \( \text{DISJ}_m \) 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 \( A = \text{diag}(A(a_1), \ldots, A(a_m)) \), Bob builds a matrix \( B = \text{diag}(B(b_1), \ldots, B(b_m)) \). It is easy to see that \( A + B \) is a permutation matrix iff \( \text{DISJ}_m(a,b) = 1 \).
Let us describe the reduction of $\text{DISJ}_m$ to $\oplus \text{PM}^{n+2}_n$. Alice and Bob first construct matrices $X_0$ and $Y_0$ of the following form:

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

then

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

where $A + B$ is a permutation matrix if $\text{DISJ}_m(a, b) = 1$. Then if $\mathcal{P}(X_0, Y_0)$ returns two cells that do not belong to the column $n$ we may conclude that $\text{DISJ}_m(a, b) = 0$. If $\mathcal{P}(X_0, Y_0)$ returns two cells from the $n$th column, then the value of $\text{DISJ}_m(a, b)$ 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 $\oplus \text{PM}^{n+2}_n$.

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

We are going to construct a protocol $\mathcal{T}$ for $\text{DISJ}_m$ 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 $X_0^{\pi, \tau}$ and $Y_0^{\pi, \tau}$ from $\mathbb{F}_2^{n+2\times n}$ such that for each $i \in [n+2]$ and $j \in [n]$, $(X_0^{\pi, \tau})_{i,j} = (X_0)_{\tau(i), \pi(j)}$ and $(Y_0^{\pi, \tau})_{i,j} = (Y_0)_{\tau(i), \pi(j)}$. Alice and Bob run the protocol $\mathcal{P}$ for inputs $X = X_0^{\pi, \tau} + \Delta$, $Y = Y_0^{\pi, \tau} + \Delta$. Notice that $X + Y = X_0^{\pi, \tau} + Y_0^{\pi, \tau}$, 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 $\bot$, then Alice and Bob return $\bot$.

First notice that if $\text{DISJ}_m(a, b) = 1$, then $\mathcal{T}$ returns a correct answer or $\bot$ with probability 1 (and the probability of $\bot$ is at most $\frac{1}{100}$), since in that case $X + Y$ has exactly one column with three cells, each of the other columns and rows contains exactly one 1-cell. Let us fix $a, b \in \{0, 1\}^m$ such that $\text{DISJ}_m(a, b) = 0$. We denote $p := \Pr[\mathcal{T}(a, b) = 0]$, we will show that $p \geq \frac{99}{250}$. 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 $\bot$, we return $\bot$, 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 $\pi$, $\tau$, and $\Delta$. Since $\text{DISJ}_m(a, b) = 0$, 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 $\text{col}(j)$ for $j \in [n]$ the set of all tuples $(x, y, j, y, j) \in ([n+2] \times [n])^2$.

$$p = \Pr_{\pi, \tau, \Delta, r} [\mathcal{P}_r(X, Y) \not\in \text{col}(\pi(n))] = \Pr_{\pi, \tau, \Delta, r} [\mathcal{P}_r(X, Y) = \bot] = 1 - \Pr_{\pi, \tau, \Delta, r} [\mathcal{P}_r(X, Y) \in \text{col}(\pi(n))] = 1 - \sum_{\pi_0, \tau_0} \Pr_{\pi, \tau, \Delta, r} [\mathcal{P}_r(X_0^{\pi_0, \tau_0} + \Delta, Y_0^{\tau_0, \pi_0} + \Delta) \in \text{col}(\pi_0(n))] \Pr_{\pi, \tau} [\pi = \pi_0, \tau = \tau_0] - p_\bot$$
Observe that for fixed $\pi_0$ and $\tau_0$ the random variable $(X_0^{\pi_0,\tau_0} + \Delta, Y_0^{\pi_0,\tau_0} + \Delta)$ is uniformly distributed over the pairs of matrices with the sum $X_0^{\pi_0,\tau_0} + Y_0^{\pi_0,\tau_0}$. Let $\alpha \in S_n$ be the transposition swapping $n$ and $4(i + 1)$ and $\beta \in S_{n+2}$ 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. $\beta$ is a product of three transpositions). By the construction of $\alpha$ and $\beta$, $(X_0 + Y_0) = (X_0^{\beta,\alpha} + Y_0^{\beta,\alpha})$, thus $(X_0^{\pi,\tau} + Y_0^{\pi,\tau}) = (X_0^{\pi_0,\tau_0} + Y_0^{\pi_0,\tau_0})$ for every $\pi, \tau$. Thus the random variable $(X_0^{\pi_0,\tau_0} + \Delta, Y_0^{\pi_0,\tau_0} + \Delta)$ has the same distribution with $(X_0^{\pi,\tau} + \Delta, Y_0^{\pi,\tau} + \Delta)$, thus we can continue the sequence as follows:

$$
p = 1 - \sum_{\pi_0,\tau_0} \Pr_{r_1}[\mathcal{P}_r(X_0^{\pi_0,\tau_0} + \Delta, Y_0^{\pi_0,\tau_0} + \Delta) \in \col((\pi_0 \circ \alpha^{-1})(n))] \Pr_{\pi,\tau}[\pi = \pi_0, \tau = \tau_0] - p_\perp
$$

$$= 1 - \sum_{\pi_0,\tau_0} \Pr_{r_1}[\mathcal{P}_r((X,Y) \in col((\pi \circ \alpha^{-1})(n)))] \Pr_{\pi,\tau}[\pi = \pi_0, \tau = \tau_0] - p_\perp
$$

$$= 1 - \Pr_{\pi,\tau}[\mathcal{P}_r(X,Y) \notin \col((\pi_0 \circ \alpha^{-1})(n))] - p_\perp
$$

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 $\ell$ and $k$ be natural numbers such that $2 \leq k \leq \ell - 7$. Then

$$R_{pub}^{1/3}(\oplus_k \text{Search}(\text{BPHP}^{2\ell+4}_2)) = \Omega\left(\frac{2^{\ell/2}}{k^{3\ell/7}}\right).$$

For $k = 2$ the stronger bound holds: $R_{pub}^{1/3}(\oplus_2 \text{Search}(\text{BPHP}^{2\ell+4}_2)) = \Omega(2^\ell)$.

We consider a combinatorial analogue of the communication problem $\oplus_k \text{Search}(\text{BPHP}^{m}_{2^\ell})$. Assume that each of $k$ parties gets $m$ binary strings from $\{0,1\}^{\ell}$, where $m > 2^\ell$. The $i$th 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 $i, s \in [m]$ such that $x_i = x_s$. We denote this problem by $\oplus_k \text{BPHP}^{m}_{2^\ell}$. It is straightforward that $R_{pub}^{1/3}(\oplus_k \text{Search}(\text{BPHP}^{m}_{2^\ell})) \geq R_{pub}^{1/3}(\oplus_k \text{BPHP}^{m}_{2^\ell})$, hence it is sufficient to prove a lower bound on $R_{pub}^{1/3}(\oplus_k \text{BPHP}^{m}_{2^\ell})$.

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

$$R_{pub}^{1/3}(\oplus_k \text{BPHP}^{2\ell+4}_2) = \Omega\left(R_{pub}^{1/3}(\text{UDISJ}_{k,2^{\ell-k-1}}) - \ell\right).$$

**Corollary 19.** $R_{pub}^{1/3}(\oplus_k \text{BPHP}^{2\ell+4}_2) = \Omega\left(\frac{2^{\ell/2}}{k^{3\ell/7}}\right)$. For $k = 2$ the stronger bound holds: $R_{pub}^{1/3}(\oplus_2 \text{BPHP}^{2\ell+4}_2) = \Omega(2^\ell)$.

**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 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].

\textbf{Proposition 20.} \(R_{pub}^{1/3}(\text{Distinct}_k,\ell) \geq R_{pub}^{1/3}(\text{UDISJ}_{k,2^{\ell-k}})\).

Let \(S_k\) denote the set of matrices from \(\{0,1\}^{2^\ell \times \ell}\) with all distinct rows. Let \(K_k \in \{0,1\}^{2^\ell \times \ell}\) be a matrix such that its \(i\)th row equals \(\text{bin}_k(i - 1 - (i - 1 \mod 2))\), i.e. the rows of \(K_k\) are \(\text{bin}_k(0), \text{bin}_k(0), \text{bin}_k(2), \text{bin}_k(2), \ldots, \text{bin}_k(2^{\ell - 1} - 2), \text{bin}_k(2^{\ell - 1} - 2)\). 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.

\textbf{Lemma 21.} There exist matrices \(A_1(0), A_1(1), \ldots, A_k(0), A_k(1) \in \mathbb{F}_2^{2^\ell \times \ell}\) 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 \(\sum_{i=1}^k A_i(b_i) \in S_k\).

\textbf{Proof of Proposition 20.} Let \((x_{i,1}, \ldots, x_{i,2^{\ell-k}})\) be an input of the \(i\)th 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 \(i\)th 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 \(t \in [1, \ldots, 2^{k} - k]\) be a matrix of size \(2^k \times (\ell - k)\) such that all its rows are equal to \(\text{bin}_{t-k}(\ell - 1)\).

Let us define \(D_1 := \begin{pmatrix} J_1 & A_1(x_{1,1}) & \cdots & \cdots & \cdots \\ J_2 & A_1(x_{1,j}) & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ J_{2^{k} - k} & A_1(x_{1,2^{k} - k}) & \cdots & \cdots & \cdots \\ \end{pmatrix} \) ; \(D_i := \begin{pmatrix} 0_{2^k \times (\ell - k)} & A_i(x_{i,1}) & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \\ 0_{2^k \times (\ell - k)} & A_i(x_{i,j}) & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \\ 0_{2^k \times (\ell - k)} & A_i(x_{i,2^{k} - k}) & \cdots & \cdots \\ \end{pmatrix} \) for \(i \in \{2, \ldots, k\}\).

By Lemma 21, the matrix \(D_1 + D_2 + \cdots + D_k\) has the following property: for all \(j \in [2^{k} - k]\), 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 \(\text{UDISJ}_{k,2^{\ell-k}}\) to \(\oplus_k \text{BPHD}_{2^\ell}^{2^k + 2^k}\) 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.

\textbf{Proof of Theorem 18.} Let \(N > 2^\ell\), consider a \(k\)-party communication problem \(\text{ROW} \oplus_k \text{BPHD}_{2^\ell}^N\), where \(i\)th 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 \(\oplus_k \text{BPHD}_{2^\ell}^N\) is that we are looking for values of a repeated row rather than numbers of equal rows.
**Claim 22.** If $R_{1/3} \left( \oplus_k \text{BPHP}_{2^k}^{N_k} \right) \leq t$, then there exists a communication protocol $P$ for $\text{ROW} \oplus_k \text{BPHP}_{2^k}^{N_k}$ using $O(t + \ell)$ bits of communication such that $P$ either returns the correct answer or $\perp$ (failure) and $\Pr[|P(M_1, \ldots, M_k)| = \perp] \leq \frac{1}{100}$ for all input matrices $M_i, i \in [k]$.

**Proof.** $P$ executes a randomized protocol for $\oplus_k \text{BPHP}_{2^k}^{N_k}$ and verifies its answer by transferring additional $O(\ell)$ bits. The probability of failure can be reduced by repetition. ⊳

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

Let $x_1, \ldots, x_k \in \{0, 1\}^{2^{\ell-k}-1}$ be inputs of the communication problem $\text{UDISJ}_{k,2^{\ell-k}-1}$. Let $x_{i,j}$ denote the $j$th bit of $x_i$ for $i \in [k], j \in [2^{\ell-k}-1]$. Let $\mathcal{P} = (x_1, x_2, \ldots, x_k)$.

### Important matrices

Let $\gamma$ be a bijection from $[2^{\ell-k} - 1] \cup \{\ast\}$ to $\{0, 1\}^{\ell-k}$, we define $k$ matrices $D_1(x_1, \gamma)$ and $D_2(x_2), D_3(x_3), \ldots, D_k(x_k)$ 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 $S_k$.

We define

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

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

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

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 $H_\mathcal{P}(\gamma) := D_1(x_1, \gamma) + D_2(x_2) + \cdots + D(x_k)$. By Lemma 21 the matrix $H_\mathcal{P}(\gamma)$ satisfies the following key property w.r.t. $(\gamma, \mathcal{P})$ in the standard basis:

**Definition 23.** Let $M$ be a matrix from $F_2^{(2^k + 2^\ell) \times \ell}$, $\gamma$ be a bijection from $[2^{\ell-k} - 1] \cup \{\ast\}$ to $\{0, 1\}^{\ell-k}$ and $e_1, e_2, \ldots, e_\ell$ be a basis in $F_\ell$.

We say that $M$ satisfies the key property w.r.t. $(\gamma, \mathcal{P})$ in the basis $(e_1, e_2, \ldots, e_\ell)$ 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 $(e_1, e_2, \ldots, e_\ell)$ are $\gamma(*)_1, \ldots, \gamma(*)_{\ell-k}$;
- $s$ appears in $M$ exactly twice.

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

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

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

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

We define random matrices $P_1, P_2, \ldots, P_k$ as follows: $P_i = M_\pi \cdot D_i(x_i) \cdot E + \Delta_i$ for $i \geq 2$ and $P_1 = M_\pi \cdot D_1(x_1) \cdot E + \Delta_1$.

- The addition of $\Delta_1$ makes $P_i$ indistinguishable from the random matrix for every $i \in [k]$.
- $\sum_{i=1}^k P_i = M_\pi C_\mathcal{P}(\gamma, E)$ and this matrix is obtained from $C_\mathcal{P}(\gamma, E)$ by the permutation $\pi$ applied to its rows.

Recall that $\mathcal{P}$ is the protocol for ROW $\oplus_k$ BPHP$^{2^\ell + 2^k}$ from Claim 22. Let $N$ be a constant to be chosen later. The protocol $\mathcal{T}$ solving UDISJ$^{k, 2^\ell + 2^k}$ 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 $O(t + \ell)$ bits. Assume that $x_1, x_2, \ldots, x_k$ is a 1-instance of UDISJ$^{k, 2^\ell + 2^k}$. Then by the key property of $C_\mathcal{P}(\gamma, E)$ all repeated rows of $\sum_{i=1}^k P_i$ are in Fake($\gamma, E$), hence the protocol $\mathcal{T}$ returns either $\bot$ or the correct answer. Since $\mathcal{P}$ is executed $N$ times independently, the probability that $Z = \{\bot\}$ is at most $\frac{1}{100^{\ell-k}}$, hence $\mathcal{T}$ returns 1 with probability at least $1 - \frac{1}{100^{\ell-k}}$.

The rest of the proof is devoted to the analysis of the case, where $x_1, x_2, \ldots, x_k$ is a 0-instance of 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
We would like to construct two bijections \(\alpha, \beta\). The random variable \(\gamma, \varepsilon\) is precisely we will define two bijections \(\alpha, \beta\) but only on the part of bits corresponding to sampling of \(\gamma, \varepsilon\). More precisely we will define two bijections \(\alpha, \beta\) on the set of values of the random variable \((\gamma, \varepsilon)\). We relax the first property as follows:

1. For every \(\gamma, \varepsilon\) the three conditional distributions of the random variable \((P_1, \ldots, P_k)\) under the following three conditions coincide:

\[
\begin{align*}
\Pr\left[\text{Fake}(\gamma, \varepsilon) \in \Fake(\gamma, \varepsilon) \cap \Fake(\varepsilon) \cap \Fake(\gamma, \varepsilon) = \emptyset\right] \leq \frac{1}{2},
\end{align*}
\]

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

1. For every \(\gamma, \varepsilon\) the three conditional distributions of the random variable \((P_1, \ldots, P_k)\) under the following three conditions coincide:

\[
\begin{align*}
\Pr\left[\text{Fake}(\gamma, \varepsilon) \in \Fake(\gamma, \varepsilon) \cap \Fake(\varepsilon) \cap \Fake(\gamma, \varepsilon) = \emptyset\right] \leq \frac{1}{2},
\end{align*}
\]
a. \((\gamma, E) = (\gamma, E)\),
b. \((\gamma, E) = \alpha(\gamma, E)\) and
c. \((\gamma, E) = \beta(\gamma, E)\).

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

The construction of \(\Xi, \alpha\) and \(\beta\) is given in Subsection 5.3.

\begin{itemize}
  \item \textbf{Definition 24.} Let \(x_1, \ldots, x_k\) be a 0-instance of \(\text{UDISJ}_{k,2^k-1}\) and \(1 > \delta \geq 0\) be an arbitrary constant. Let \(\Xi\) be a set consisting of pairs \((\gamma, E)\), where \(\gamma\) is a bijection from \([2^k-1]\) to \((0,1)^{2^k-1}\), \(E\) is an invertible matrix from \(F_{2^k}^{2^k}\). Let \(\alpha\) and \(\beta\) be bijections from \(\Xi\) to \(\Xi\). We say that \((\Xi, \alpha, \beta)\) forms a \((1 - \delta)\)-symmetry randomness space for \(\mathcal{P}\) if the following conditions hold:
    \begin{itemize}
      \item \textbf{(Largeness)} \(\Pr[\gamma(\mathcal{P}) \in \Xi] \geq 1 - \delta\).
      \item \textbf{(Difference)} For all \((\gamma, E) \in \Xi\), \(\text{Fake}(\gamma, E) \cap \text{Fake}(\alpha(\gamma, E)) \cap \text{Fake}(\beta(\gamma, E)) = \emptyset\).
      \item \textbf{(Symmetry)} For all \((\gamma, E) \in \Xi\) the matrices \(C_{\mathcal{F}}(\gamma, E), C_{\mathcal{F}}(\alpha(\gamma, E))\) and \(C_{\mathcal{F}}(\beta(\gamma, E))\) differ only by a permutation of rows.
    \end{itemize}

\begin{itemize}
  \item \textbf{Claim 25.} Assume that \(x_1, \ldots, x_k\) is a 0-instance of \(\text{UDISJ}_{k,2^k-1}\), \(1 > \delta \geq 0\) is a constant. Let \((\Xi, \alpha, \beta)\) form a \((1 - \delta)\)-symmetry randomness space for \(\mathcal{P}\).

Then

\[
\Pr[\mathcal{P}(P_1, P_2, \ldots, P_k) \in \text{Fake}(\gamma, E)] \leq \frac{2}{3} + \delta.
\]

Proof. Let us denote \(\vec{P} = (P_1, P_2, \ldots, P_k), \Delta = (\Delta_1, \Delta_2, \ldots, \Delta_k)\) and \(\vec{D}(\mathcal{F}, \gamma) = (D_1(x_1, \gamma), D_2(x_2), \ldots, D_k(x_k))\).

\[
\vec{P} = (\Delta_1 + M_\pi D_1(x_1, \gamma)E, \Delta_2 + M_\pi D_2(x_2)E, \ldots, \Delta_k + M_\pi D_k(x_k)E),
\]

for brevity we use the vector notation \(\vec{P} = \vec{\Delta} = \Delta + M_\pi \vec{D} = \Delta + M_\pi \vec{D}(\mathcal{F}, \gamma)\).

Let \(p := \Pr[\mathcal{P}(\vec{P}) \in \text{Fake}(\gamma, E)]\).

\[
p = \sum_{\gamma, E} \Pr[\mathcal{P}(\vec{\Delta} + M_\pi \vec{D}(\mathcal{F}, \gamma) \cdot E) \in \text{Fake}(\gamma, E)] \cdot \Pr[\gamma = \gamma, E = E]
\]

\[
\leq \sum_{(\gamma, E) \in \Xi} \Pr[\mathcal{P}(\vec{\Delta} + M_\pi \vec{D}(\mathcal{F}, \gamma) \cdot E) \in \text{Fake}(\gamma, E)] \cdot \Pr[\gamma = \gamma, E = E] + \delta
\]

Notice that for fixed \(\gamma, E\) the random variable \(\vec{\Delta} + M_\pi \vec{D}(\mathcal{F}, \gamma) \cdot E\) is distributed uniformly on the set of tuples \(\sum_{i=1}^k L_i\) of \(k\) matrices from \(F_{2^k}^{(2^k+2^i) \times 2^i}\) such that \(\sum_{i=1}^k L_i\) differs from \(C_{\mathcal{F}}(\gamma, E)\) only by a permutation of rows. Let \((\gamma_\alpha^{-1}, E_\alpha^{-1}) = \alpha^{-1}(\gamma, E)\). By the symmetry condition, matrices \(C_{\mathcal{F}}(\gamma, E)\) and \(C_{\mathcal{F}}(\gamma_\alpha^{-1}, E_\alpha^{-1})\) differ only by permutation of rows. Thus, for every set \(A\) the probability \(\Pr[\mathcal{P}(\vec{\Delta} + M_\pi \vec{D}(\mathcal{F}, \gamma_\alpha^{-1}) \cdot E) \in A]\) = \(\Pr[\mathcal{P}(\vec{\Delta} + M_\pi \vec{D}(\mathcal{F}, \gamma_{\alpha^{-1}}) \cdot E) \in A]\). Hence,
Proof of Lemma 26. Assume that
\[ p \leq \sum_{(\gamma, E) \in \Xi} \Pr\left[ P \left( \overrightarrow{\Delta} + M_{\star} \cdot \left( \overrightarrow{D}(\overrightarrow{\mathcal{F}}, \gamma_{a-1}) \cdot E_{a-1} \right) \right) \in \text{Fake}(\gamma, E) \right] \cdot \Pr[\gamma = \gamma, E = E] + \delta \]
\[ = \sum_{(\gamma, E) \in \Xi} \Pr\left[ P \left( \overrightarrow{\Delta} + M_{\star} \cdot \left( \overrightarrow{D}(\overrightarrow{\mathcal{F}}, \gamma) \cdot E \right) \right) \in \text{Fake}(\alpha(\gamma, E)) \right] \cdot \Pr[(\gamma, E) = \alpha(\gamma, E)] + \delta \]
\[ = \sum_{(\gamma, E) \in \Xi} \Pr\left[ P \left( \overrightarrow{\Delta} + M_{\star} \cdot \left( \overrightarrow{D}(\overrightarrow{\mathcal{F}}, \gamma) \cdot E \right) \right) \in \text{Fake}(\alpha(\gamma, E)) \right] \cdot \Pr[(\gamma, E) = (\gamma, E)] + \delta \]
\[ = \Pr\left[ P \left( \overrightarrow{P} \right) \in \text{Fake}(\alpha(\gamma, E)), (\gamma, E) \in \Xi \right] + \delta. \]

Analogously, \( p \leq \Pr\left[ P \left( \overrightarrow{P} \right) \in \text{Fake}(\beta(\gamma, E)), (\gamma, E) \in \Xi \right] + \delta. \) Also the inequality \( p \leq \Pr\left[ P \left( \overrightarrow{P} \right) \in \text{Fake}(\gamma, E), (\gamma, E) \in \Xi \right] + \delta \) follows by the largeness condition. Then,
\[ 3(1 - p) \geq \Pr\left[ P \left( \overrightarrow{P} \right) \not\in \text{Fake}(\beta(\gamma, E)) \lor (\gamma, E) \not\in \Xi \right] \]
\[ + \Pr\left[ P \left( \overrightarrow{P} \right) \not\in \text{Fake}(\alpha(\gamma, E)) \lor (\gamma, E) \not\in \Xi \right] \]
\[ + \Pr\left[ P \left( \overrightarrow{P} \right) \not\in \text{Fake}(\gamma, E) \lor (\gamma, E) \not\in \Xi \right] - 3\delta \]
\[ \geq \Pr\left[ P \left( \overrightarrow{P} \right) \not\in \text{Fake}(\gamma, E) \cap \text{Fake}(\alpha(\gamma, E)) \cap \text{Fake}(\beta(\gamma, E)) \lor (\gamma, E) \not\in \Xi \right] - 3\delta \]
\[ = 1 - 3\delta. \]

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 UDISH\(_{k,2^{k-1}+1}\). Then for some \( \delta < \frac{1}{3} - \frac{1}{100} \) there exists a \( (1 - \delta) \)-symmetry randomness space for \( \overrightarrow{\mathcal{F}} \).

Lemma 26 and Claim 25 imply that there is a constant \( \varepsilon > 0 \) such that
\[ \Pr\left[ P \left( P_1, P_2, \ldots, P_k \right) \not\in \text{Fake}(\gamma, E) \right] \geq \varepsilon + \frac{1}{100}. \]

Thus,
\[ \Pr\left[ P \left( P_1, P_2, \ldots, P_k \right) \not\in \text{Fake}(\gamma, E) \cup \{ \perp \} \right] \geq \varepsilon. \]

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

### 5.3 Constructions of \( \Xi, \alpha \) and \( \beta \)

**Proof of Lemma 26.** Assume that \( x_1, \ldots, x_k \) is a 0-instance UDISH\(_{k,2^{k-1}+1}\). Let \( i_0 \in [2^{k-1} - 1] \) be such that \( x_{i_0} = x_{2,i_0} = \ldots = x_{k,i_0} = 1 \).

Hereinafter \( \gamma \) denotes a bijection from \( [2^{k-1} - 1] \cup \{ \ast \} \) to \( \{0, 1\}^{\ell-k} \), \( E \) denotes an invertible matrix from \( \mathbb{F}_2^{\ell \times T} \) and \( e_1, e_2, \ldots, e_T \) denote rows of \( E \).

Before presenting constructions of \( \Xi, \alpha \), and \( \beta \) 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:
\[ R(s, b, E) := \{ (s, b, z) \cdot E \mid z \in \mathbb{F}_2^{k-1} \}. \]
Using the key property of the matrix $C_\mathcal{P}(\gamma, E)$ we can describe rows of $C_\mathcal{P}(\gamma, E)$ in terms of $R(s, b, E)$.

▷ Claim 27.
- The set of the last $2^k+1$ rows of $C_\mathcal{P}(\gamma, E)$ 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 Fake$(\gamma, E)$. Hence, Fake$(\gamma, E) = R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$.
- The set of rows of $C_\mathcal{P}(\gamma, E)$ 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 $C_\mathcal{P}(\gamma, E)$ with indices from $[2^k i_0 - 1; 2^k i_0]$ 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 $\text{Span}(e_{\ell-k+2}, \ldots, e_{\ell})$:

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

Proof.

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

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

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

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

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

Proof. By Claim 27, rows from $R(\gamma(i_0), 1, E)$ do not appear in $C_\mathcal{P}(\gamma, E)$, rows from $R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ appear in $C_\mathcal{P}(\gamma, E)$ exactly twice. The matrix $C_\mathcal{P}(\gamma, E)$ has $2^k + 2^{k'}$ rows. All rows of $C_\mathcal{P}(\gamma, E)$ 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 $C_\mathcal{P}(\gamma, E)$ exactly once.
By Claims 29 and 30, \( |R(\gamma(i_0), 0, E) \cup R(\gamma(\ast), 0, E) \cup R(\gamma(\ast), 1, E)| = 3 \cdot 2^{k-1} \), hence, the number of rows of \( C_\gamma(E) \) that are not in \( R(\gamma(i_0), 1, E) \cup R(\gamma(\ast), 0, E) \cup R(\gamma(\ast), 1, E) \) equals \( 2^k - 2^{k+1} \). By Claims 29 and 30, the number of \( \ell \)-hit strings not from \( R(\gamma(i_0), 1, E) \cup R(\gamma(\ast), 0, E) \cup R(\gamma(\ast), 1, E) \) is also \( 2^k - 2^{k+1} \). Hence, all rows from \( \{0, 1\}^\ell \setminus (R(\gamma(i_0), 0, E) \cup R(\gamma(\ast), 0, E) \cup R(\gamma(\ast), 1, E) \cup R(\gamma(i_0), 1, E)) \) appear in \( C_\gamma(E) \) exactly once. Thus, matrices \( C_{\gamma'}(E, \gamma) \) and \( C_{\gamma'}(E', \gamma') \) have the same set of rows and each row appears the same number of times in each of these matrices.

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 \( \alpha \) and \( \beta \) such that for all \( (\gamma, E), \xi \in \Xi \) the following equalities are satisfied.

\[
\begin{aligned}
R(\gamma(i_0), 1, E) &= R(\gamma_\alpha(i_0), 1, E_\alpha) = R(\gamma_\beta(i_0), 1, E_\beta); \\
R(\gamma(i_0), 0, E) &= R(\gamma_\alpha(\ast), 0, E_\alpha) = R(\gamma_\beta(\ast), 0, E_\beta); \\
R(\gamma(\ast), 1, E) &= R(\gamma_\alpha(\ast), 1, E_\alpha) = R(\gamma_\beta(\ast), 1, E_\beta); \\
R(\gamma(\ast), 0, E) &= R(\gamma_\alpha(\ast), 0, E_\alpha) = R(\gamma_\beta(\ast), 1, E_\beta).
\end{aligned}
\]  

(2)

Notice that by Claim 31, equations (2) imply the symmetry property. Equations (2) also imply the difference property. Indeed,
- \( \text{Fake}(\gamma, E) = R(\gamma(\ast), 1, E) \cup R(\gamma(\ast), 0, E) \);
- \( \text{Fake}(\gamma_\alpha, E_\alpha) = R(\gamma_\alpha(\ast), 1, E_\alpha) \cup R(\gamma_\alpha(\ast), 0, E_\alpha) = R(\gamma(\ast), 1, E) \cup R(\gamma(i_0), 0, E) \);
- \( \text{Fake}(\gamma_\beta, E_\beta) = R(\gamma_\beta(\ast), 1, E_\beta) \cup R(\gamma_\beta(\ast), 0, E_\beta) = R(\gamma(\ast), 0, E) \cup R(\gamma(i_0), 0, E) \).

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

In order to complete the proof of the lemma we have to construct \( \Xi \) and bijections \( \alpha, \beta \) from \( \Xi \) to \( \Xi \) such that
- \( \text{(Largeness)} \quad \Pr[(\gamma, E) \in \Xi] > \frac{2}{3} + \frac{1}{100}; \)
- and for all \( (\gamma, E), \xi \in \Xi \) the equations (2) are satisfied.

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

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

\[
\Pr[(\gamma, E) \in \Xi] = 1 - \Pr[\gamma(i_0) \leq \gamma(\ast) \lor \gamma(\ast) \leq \gamma(i_0)] \geq 1 - 2 \Pr[\gamma(i_0) \leq \gamma(\ast)]
= 1 - 2 \Pr[S \subseteq T | S \not\subseteq T] \geq 1 - 2 \Pr[S \subseteq T]
= 1 - 2 \prod_{j=1}^{\ell-k} (1 - \Pr[j \in S \land j \not\in T]) = 1 - 2 \prod_{j=1}^{\ell-k} (1 - \frac{3}{4}) = \frac{2}{3} + \frac{1}{100} \text{ if } \ell - k \geq 7.
\]

Hence, the largeness property is satisfied.

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

\[
\gamma_\alpha(i) = \begin{cases} 
\gamma(i) & \text{if } i = i_0 \\
\gamma(\ast) & \text{if } i = i_0 \\
\gamma(i) & \text{otherwise}
\end{cases}.
\]
Thus, we can uniquely recover \( \gamma \) since \( \sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j \in \text{Span}(e_1,\ldots,e_{\ell-k}) \). 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 \( \alpha \), we get that \( \alpha(\Xi) \subseteq \Xi \). Notice that \( \sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j = \sum_{j=1}^{\ell-k}(\gamma(0)_j + \gamma(*)_j)e'_j \), hence \( \alpha(\gamma, E_\alpha) = (\gamma, E) \), hence \( \alpha \) is bijective.

\[ \sum_{k=1}^{\ell-k} \gamma(i_0)_k e_k e_j + e'_{k+1} = R(\gamma(i_0), 1, E); \]
\[ R(\gamma(0), 0, E) = R(\gamma(0), 0, E); \]
\[ R(\gamma(0), 0, E) = R(\gamma(0), 0, E); \]
\[ R(\gamma(0), 1, E) = R(\gamma(0), 1, E). \]

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

\[ R(\gamma(0), 1, E) = \left( \sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j + e'_{k+1} \right) + \sum_{j=1}^{\ell-k} \gamma(0)_j e^*_j + e'_{k+1} + e^*_j + e'_{k+1} = R(\gamma(i_0), 1, E); \]
\[ R(\gamma(0), 0, E) = \left( \sum_{j=1}^{\ell-k} \gamma(0)_j e^*_j + e'_{k+1} \right) + \sum_{j=1}^{\ell-k} \gamma(0)_j e^*_j + e'_{k+1} + e^*_j + e'_{k+1} = R(\gamma(0), 0, E); \]
\[ R(\gamma(0), 1, E) = \left( \sum_{j=1}^{\ell-k} \gamma(0)_j e^*_j + e'_{k+1} \right) + \sum_{j=1}^{\ell-k} \gamma(0)_j e^*_j + e'_{k+1} + e^*_j + e'_{k+1} = R(\gamma(0), 1, E). \]

Construction of \( \beta \). For \( (\gamma, E) \in \Xi \), we define \( \beta(\gamma, E) = (\gamma, E_\beta) \), where \( \gamma_\beta = \gamma \alpha \) and \( E_\beta \) 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 \( E_\beta = (e^\prime,\ldots,e^\prime) \):

\[ e^\prime_j = \begin{cases} e_j & \text{if } j \notin \{ j_{\min}, \ell - k + 1 \} \\ \sum_{i=1}^{j_{\min}} (\gamma(0)_i + \gamma(i_0)_i) e_i & \text{if } j = \ell - k + 1 \\ e_{j_{\min}} + e'_{k+1} & \text{if } j = j_{\min} \end{cases}. \]

Proof. Let us verify that \( \beta \) is injective. Given \( \gamma_\beta \) we can easily recover \( \gamma \), hence we can recover \( j_{\min} \) as well. Then

\[ \sum_{i=1}^{\ell-k} (\gamma(i_0)_i + \gamma(0)_i) e_i + e'_{\ell-k+1} = \sum_{i \in [\ell-k] \setminus \{ j_{\min} \}} (\gamma(i_0)_i + \gamma(0)_i) e_i + e_{j_{\min}} + e'_{\ell-k+1} + e'_{\ell-k+1} = e_{\ell-k+1} + \sum_{i \in [\ell-k] \setminus \{ j_{\min} \}} (\gamma(i_0)_i + \gamma(0)_i) e_i + e_{j_{\min}} + e'_{\ell-k+1} = e_{\ell-k+1}. \]

Thus, we can uniquely recover \( e_{\ell-k+1} \) and, hence, also recover \( e_{j_{\min}} = e'_{j_{\min}} + e_{\ell-k+1} \); for \( j \in [\ell] \setminus \{ j_{\min}, \ell - k + 1 \} \), \( e_j = e^\prime_j \). Hence, \( \beta \) is injective. Notice that since we represent \( e_1,\ldots,e_\ell \) as linear combinations of \( e^\prime,\ldots,e^\prime \), then \( e^\prime,\ldots,e^\prime \) is a basis, hence the matrix \( E_\beta \) is invertible. Thus, we verify that \( \beta(\Xi) \subseteq \Xi \) and \( \beta \) is injective, hence \( \beta \) is bijective.
Proof. We denote \( S := \text{Span}(e_{\ell-k+2}, \ldots, e_k) = \text{Span}(e_{\ell-k+2}', \ldots, e_k') \). Recall that \( \gamma(*)_{j_{\min}} = 1 \) and \( \gamma(i_0)_{j_{\min}} = 0 \).

1. \( R(\gamma_0(i_0), 1, E_0) = R(\gamma_0(i_0), 1, E) \);
2. \( R(\gamma_0(i_0), 0, E_0) = R(\gamma(*), 1, E) \);
3. \( R(\gamma_0(*), 0, E_0) = R(\gamma(i_0), 0, E) \);
4. \( R(\gamma_0(*), 1, E_0) = R(\gamma(*), 0, E) \);

Proof of Lemma 21. Let us verify that all conditions hold. First we show that the matrices \( A_i(0) = T_i \) and \( A_i(1) \) be the zero matrix. Let \( A_i(0) = K_k + T_i \) and \( A_i(1) = K_k \). For each \( b_1, \ldots, b_k \in \{0, 1\} \), \( \sum_{i=1}^k A_i(b_i) = \sum_{i=1}^k (1 - b_i)T_i + K_k \). 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 \( M \in \text{Span}(T_1, \ldots, T_k) \) 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 \( K_{k+1} = \begin{pmatrix} K_k & 0_{2^k \times 1} \\ K_k & 1_{2^k \times 1} \end{pmatrix} \). Let \( T_1, \ldots, T_k \) be the matrices from induction hypothesis for \( k \). Then define \( T'_i = \begin{pmatrix} T_i & 0_{2^k \times 1} \\ T_i & 1_{2^k \times 1} \end{pmatrix} \) for \( i \in \{k\} \) and \( T'_{k+1} = \begin{pmatrix} 0_{2^k \times k} & z_0 \\ 1_{2^k \times k} & z_1 \end{pmatrix} \), where \( z_0 = (0, 1, 0, 1, \ldots, 0, 1)^T \in \{0, 1\}^{2^k \times 1} \) and \( z_1 = (1, 0, 1, 0, 1, \ldots, 0, 1)^T \in \{0, 1\}^{2^k \times 1} \).

Let us verify that all conditions hold. First we show that the matrices \( T'_1, T'_2, \ldots, T'_{k+1} \) are linearly independent. Matrices \( T'_1, T'_2, \ldots, T'_k \) are linearly independent since they contain linearly independent blocks \( T_1, \ldots, T_k \). The matrix \( T'_{k+1} \) does not belong to \( \text{Span}(T'_1, \ldots, T'_k) \), since the last column of \( T'_i \) is non-zero, but the last columns of \( T'_1, \ldots, T'_k \) are zeros.
Let us check that for any non-zero matrix $M \in \text{Span}(T'_1, \ldots, T'_k)$, the condition $M + K_k + 1 \in S_{k+1}$ 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 $\text{Span}(T'_1, \ldots, T'_k)$. Then, $M$ has form $(M', 0_{2^k \times 1})$, where $M'$ is a non-zero matrix from $\text{Span}(T_1, \ldots, T_{k})$, thus $M' + K_k \in S_k$. Then $M + K_k + 1 = (M' + K_k, 0_{2^k \times 1})$; it follows from the induction hypothesis that all rows of this matrix are distinct, i.e. $M + K_k + 1 \in S_{k+1}$. 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. $M = T'_k$, then $M + K_k + 1 = (K_k, 1_{2^k \times 1} + K_k - z_0)$. 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 $M + K_k + 1$ coincides with the $(2i + 2i)$th row of $M + K_k + 1$ with flipped bits, and the $(2i)$th row of $M + K_k + 1$ coincide with the $(2i + 2i - 1)$th row of $M + K_k + 1$ with flipped bits for $i \in [2^k]$.

3. $M = R + T'_k$, where $R$ is a non-zero matrix from $\text{Span}(T_1, \ldots, T_k)$. Let $R'$ have the form $(R', 0_{2^k \times 1})$, where $R'$ is a non-zero matrix from $\text{Span}(T_1, \ldots, T_k)$. Then $M + K_k + 1 = R + T'_{k+1} + K_{k+1} = (R' + K_k, 1_{2^k \times 1} + R' + K_k - z_0)$. By the induction hypothesis, $R' + K_k \in S_k$ 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, $M + K_k + 1 \in S_{k+1}$.

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 $i$th row of $R' + K_k$ coincides with the $j$th row of $R' + K_k$ with flipped bits, then the $i$th row of $M + K_k + 1$ coincides with its $j$th 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 $k + 7 \leq \ell$, then the size of any semantic $\text{Res}(\text{PC}_{k-1})$ tree-like refutation of $\text{BPHP}_{2^k + 2^k}$ is at least $2^\Theta \left( \frac{2^k}{k^{2/7} \ell} \right)$. For $k = 2$, the size of any tree-like semantic $\text{Res}(\oplus)$ refutation of $\text{BPHP}_{2^2 + 4}$ is at least $2^\Theta(2^\ell)$.

**Proof.** Follows from Theorem 3 and Lemma 1.

---

CCC 2021
Corollary 38. Let $2 \leq k \leq \ell - 7$ and $S$ be the minimal size of tree-like refutation of $\varphi = \text{BPHP}_2^{2\ell} \circ \oplus_k$ in the semantic proof system $\text{T}_{cc}(k, c)$. Then $\log S \log \log S \geq c \cdot \Omega \left( \frac{2^{\ell/2}}{k^{2\ell+c^2}} \right)$.

For $k = 2$, $\log S \log \log S \geq c \cdot \Omega (2^\ell)$.

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

$$R_{\text{pub}}^{1/3}(\text{Search} (\text{BPHP}_2^{2\ell} \circ \oplus_k)) \geq R_{\text{pub}}^{1/3} \left( \oplus_k \text{Search} (\text{BPHP}_2^{2\ell}) \right).$$

Now the statement follows from Theorem 3.

Corollary 38.

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 $f_1 : X_1 \rightarrow Z_1$, $f_2 : X_2 \rightarrow Z_2, \ldots, f_k : X_k \rightarrow Z_k$ and $g : W \rightarrow Y$ 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 $R_{1/3}^{\text{pub}}(S) \leq R_{1/3}^{\text{pub}}(T)$.

Proof. The $i$th party computes $f(x_i)$ 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.

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

Recall that BPHP$_m^\ell$ 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 $\Pi_k$ of the variables of BPHP$_m^\ell$ into $k$ parts. Let $n = \ell k + r$ where $0 \leq r < k$. For each $i \in [M]$ the row $s_i$ is partitioned into $k$ parts $s_i = s_i^{(1)} s_i^{(2)} \cdots s_i^{(k)}$ such that $|s_i^{(1)}| = \ell + 1$ if $t \leq r$, and $|s_i^{(1)}| = \ell$ if $t > r$. The partition $\Pi_k$ of the variables of BPHP$_m^\ell$ into $k$ parts is the following: the $t$th part consists of the variables $s_1^{(t)}, s_2^{(t)}, \ldots, s_M^{(t)}$.

We consider a search problem SearchPair$_{2\ell}^\ell$: given the values of the variables of BPHP$_m^\ell$, that are partitioned according to $\Pi_k$ 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 SearchPair$_{2\ell}^\ell$ is many-one reducible to Search $\left( \text{BPHP}_m^\ell \right)$ with variables partitioned according to $\Pi_k$.

Proof. The proof is straightforward.

Theorem 41. $\oplus_k \text{BPHP}_m^\ell$ is many-one reducible to SearchPair$_{2\ell^{2(k-1)/\ell}}^\ell$.

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

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

For every $t \in [k]$ we define $f_t$ as follows: $f_t(x_1^{(t)}, \ldots, x_M^{(t)})$ is a sequence of rows $r_1^{(t)}, r_2^{(t)}, \ldots, r_M^{(t)}$ such that for all $i \in [M]$, $r_i^{(t)} = z_i + x_i^{(t)}$, 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 $z_i$ denotes the $t$th part).

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

Let $q, w \in [M]$ and $q \neq w$. 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 \( f_1, f_2, \ldots, f_k \) and \( g \) define a reduction. Let \( q, w \in M \) be a pair of different numbers such that the assignment \( \alpha := \{ s_i \leftarrow r_i^{(1)} r_i^{(2)} \ldots r_i^{(k)} \mid i \in [M] \} \) 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 \( \alpha \) 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 \( \varphi \) is a bijection, we get \( q = w \). Thus, \( j_1 \neq j_2 \).

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

\[
z_t + x_{j_2}^{(t)} = y_t + x_{j_1}^{(t)} \quad (3)
\]

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 \( \oplus_k \text{BPHP}^m_{2^t} \).

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 \( 0 \leq r < k \). Let \( M > 2^k \ell \). Then SearchPair\(^M\) \( M^2 \) is many-one reducible to SearchPair\(^M\) \( 2^t \).

**Proof.** Let \( x_1, x_2, \ldots, x_M \) be the input of SearchPair\(^M\) \( 2^t \), let \( x_i^{(t)} \) be the \( t \)th part of the row \( x_i \) according to the partition \( \Pi_i \). Given this input we construct an input for SearchPair\(^M\) \( 2^t \).

Let \( \tau \) be a bijection between \([M] \times \{0, 1\}^r \) and \([M^2]^r \). For each \( i \in [M] \) we construct \( 2^r \) rows \( y_{\tau(i,\alpha)} \) one for each \( \alpha \in \{0, 1\}^r \). Let \( \Pi_k \) partition a row \( y_{\tau(i,\alpha)} \) into the following parts: \( y_{\tau(i,\alpha)}^{(1)} y_{\tau(i,\alpha)}^{(2)} \ldots y_{\tau(i,\alpha)}^{(k)} \).

Let

\[
y_{\tau(i,\alpha)}^{(t)} = \begin{cases} x_i^{(t)} & \text{if } t > r \\ x_i^{(r)} & \text{if } 0 \leq t \leq r \end{cases}
\]

Now we can define the function \( f_t(x_1^{(t)}, \ldots, x_M^{(t)}) \) 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 \( y_{\tau(i,\alpha)} \) 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 \( M = 2^n + 2^{k+n-\left\lceil n/k \right\rceil} \) and \( n \geq k(k + 7) \). If variables of BPHP\(^M\) \( 2^n \) are partitioned according \( \Pi_k \), then \( R_{1/3}^{\text{pub}} \bigl( \text{Search} \bigl( \text{BPHP}^M_{2^n} \bigr) \bigr) = \Omega \left( \frac{2^{n/2 - 3k/2}}{k} \right) \).

For \( k = 2 \) a stronger bound holds: \( R_{1/3}^{\text{pub}} \bigl( \text{Search} \bigl( \text{BPHP}^M_{2^n} \bigr) \bigr) = \Omega \left( 2^{n/2} \right) \).

**Proof.** Let \( \ell = \left\lfloor n/k \right\rfloor \) and \( r = n - \ell k \).

\[
R_{1/3}^{\text{pub}} \bigl( \text{Search} \bigl( \text{BPHP}^M_{2^n} \bigr) \bigr) = R_{1/3}^{\text{pub}} \bigl( \text{Search} \bigl( \text{BPHP}^{2^n+2^\ell} \bigr) \bigr)
\]

\[
\geq R_{1/3}^{\text{pub}} \bigl( \text{SearchPair}^{2^n+2^\ell} \bigr)
\]

\[
\geq R_{1/3}^{\text{pub}} \bigl( \text{SearchPair}^{2^n} \bigr)
\]

\[
\geq R_{1/3}^{\text{pub}} \left( \oplus_k \text{BPHP}^{2^n+2^\ell} \right) \quad \text{(Corollary 19)}
\]

\[
\geq \Omega \left( \frac{2^{\ell/2} - 3k/2}{k} \right) = \Omega \left( \frac{2^{n/2 - 3k/2}}{k} \right)
\]

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 Search \( \text{BPHP}_{2^n}^m \)

**Proposition 5.** For \( M > 2^n \) and \( k \in \{2,3,\ldots,n\} \) there exists a deterministic \( \text{NOF} \) communication protocol for Search \( \text{BPHP}_{2^n}^M \) w.r.t. \( \Pi_k \) transmitting \( O(2^{[n/k]} \cdot \log M) \) 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 \( s_i^{(1)}, \ldots, s_i^{(M)} \) and that Bob can see all other variables.

Let us denote \( s_i^{(1)} = i \in [M] \) the bits Bob sees in the \( i \)th line for \( i \in [M] \). Bob finds a value \( \alpha \in \{0,1\}^{n-[n/k]} \) such that the size of the set \( S_\alpha = \{ i \in [M] \mid s_i^{(1)} = \alpha \} \) is larger than \( 2^{[n/k]} \). Such \( \alpha \) exists since \( M > 2^n \). Bob then picks an arbitrary subset \( S' \) of \( S_\alpha \) of size \( 2^{[n/k]} + 1 \) and sends the description of \( S' \) to Alice using \( (2^{[n/k]} + 1) \cdot \log_2 M \) bits. Then, by the pigeonhole principle there exists \( i \neq j \in S' \) such that \( s_i^{(1)} = s_j^{(1)} \). Alice and Bob then spend \( O(\log M + n) \) bits transmitting indices \( i \) and \( j \) and all the values of the \( i \)th and \( j \)th lines to each other. Both of them then find the falsified clause of \( \text{BPHP}^M \) 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 \( O(n + \log M) \) bits. \( \blacktriangleleft \)

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 \( \text{Th}(\log n) \) proof of \( \text{BPHP}_{2^n}^m \)

In this section we give a short tree-like \( \text{Th}(\log n) \) refutation of the bit pigeonhole principle \( \text{BPHP}_{2^n}^m \). This observation is similar to the one of \([5]\) that converts a resolution proof of the unary encoding of the pigeonhole principle \( \text{PHP}_{2^n}^m \) to a proof of \( \text{BPHP}_{2^n}^m \) in \( \text{Res}(\log n) \).

Namely we prove the following:

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

**Proof.** Let \( p_{i,j} \) for \( i \in [m] \) and \( j \in [2^\ell] \) be a variable of \( \text{PHP}_{2^n}^m \) indicating that the \( i \)th pigeon flies to the \( j \)th hole. Let \( s_{i,k} \) for \( i \in [m], k \in [\ell] \) be a variable of \( \text{BPHP}_{2^n}^m \) indicating the \( \ell \)th bit of the \( i \)th string \( s_i \).

Let \( Q_j(x_1, x_2, \ldots, x_\ell) \) for \( j \in [2^\ell] \) 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 \( (a_1, a_2, \ldots, a_\ell) = \text{bin}_\ell(j-1) \) and \( Q_j(a_1, a_2, \ldots, a_\ell) = 0 \) otherwise. We can define \( Q_j \) as follows \( Q_j(x_1, \ldots, x_\ell) = \prod_{k=1}^\ell (1 - x_k + \alpha) \) for \( i \in [m], j \in [2^\ell] \), where \( \alpha = \text{bin}_\ell(j-1) \). By the construction \( \deg(Q_j) = \ell \).

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

Consider a tree-like \( \text{Th}(1) \)-refutation of \( \text{PHP}_{2^n}^m \) 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 \( \text{PHP}_{2^n}^m \).

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 \( \text{Th}(\ell) \) refutation of \( \text{BPHP}_{2^n}^m \).
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.

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 \( \{0, 1\}^{\text{Vars}(\text{PHP}_{2i}^m)} \) for \( a, c \in [m], b \in [2^f] \). 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 \( s_a \neq \text{bin}_f(b) \lor s_c \neq \text{bin}_f(b) \) which is an axiom of \( \text{BPHP}_{2i}^m \).

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

It is easy to see that the size of the resulting refutation is at most \( 3S \).

\[ \blacksquare \]

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

**Proposition 6.** For \( m > 2^f \) there exists a tree-like \( \text{Th}(f) \) refutation of \( \text{BPHP}_{2i}^m \) of size \( O(m^2 \cdot 2^f) \).

**Proof.** Follows from Propositions 43 and 44.

---

**References**


Proof of Lemma 12

Lemma 12. Let $T$ be a binary tree with $m$ vertices such that the $i$th 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 $A$ that with probability at least $\frac{2}{3}$ returns a leaf $\ell$ of $T$ with $a_\ell = 1$ and makes $O(\log m)$ 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 Subtree($F, v$) the subtree of $F$ with root $v$. Let Oracle($i$) be the oracle function returning the correct value of $a_i$ with probability at least $\frac{2}{3}$. 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 $3C[\log m]$ iterations we obtain the required error probability by Markov’s inequality.
Algorithm 2 Search for 1-leaf.

\[ T_0 := T \]
\[ i := 0 \]
\[ \text{for } j := 1 \text{ to } 3C[\log_{3/2} m] \text{ do} \]
\[ r := \text{root of } T_i \]
\[ \text{if Oracle}(r) = 0 \text{ then} \]
\[ i := \max\{0, i - 1\} \quad \triangleright \text{Backtrack since the current tree may not contain a 1-leaf} \]
\[ \text{else if } |T_i| \neq 1 \text{ then} \]
\[ v := \text{a centroid node of } T_i \]
\[ \text{if Oracle}(v) = 1 \text{ then} \]
\[ T_{i+1} := \text{Subtree}(T_i, v) \]
\[ \text{else} \]
\[ T_{i+1} := T_i \setminus \text{Subtree}(T_i, v) \]
\[ \triangleright T_{i+1} \text{ is obtained from } T_i \text{ by the deletion of } \text{Subtree}(T_i, v) \]
\[ i := i + 1 \]
\[ \text{return the only node of } T_i \text{ if } |T_i| = 1 \]

Let \( T(j) \) denote the value of \( T_i \) before the start of the \( j \)-th iteration, \( i(j) \) denote \( i \) at the start of the \( j \)-th iteration and \( r(j) \) denote the root of \( T(j) \). Notice that if \( a_{r(j)} = 1 \), then for every \( j' > j \), \( T(j') \) is a subtree of \( T(j) \), 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 \( j_1 = 0 \), \( j_s = \min\{j \mid a_{r(j)} = 1 \land j > j_{s-1} \land i(j) > i(j_{s-1})\} \), 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:

1. If \( T(j) \) coincides with \( T(j_s) \), 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 \( \Pr[Y_j = -1] \geq \frac{3}{10} \).
2. If the root of \( T(j) \) is labeled with zero, then \( Y_j = -1 \), if the first oracle query returns the correct answer (i.e. the algorithm backtracks). Otherwise, if \( T(j) \) consists of a single node \( Y_j = 0 \). Otherwise, if the root of \( T(j+1) \) 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, \( j_{s+1} = j_s + \min\{k \mid \sum_{j=j_s}^{j+k-1} Y_j \leq -1\} \). 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 \( \sum_{j=j_s}^{j+k-1} Y_j \leq \sum_{j=j_s}^{j+k-1} X_j \). 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 \( |T_{j_0}| \leq \left( \frac{3}{4} \right)^{m} |T_{j_0}| \), the algorithm that runs for \( 3C[\log_{3/2} m] \) iterations terminates in a 1-labeled leaf with probability at least \( \frac{2}{3} \).