Read-Once Branching Programs, Rectangular Proofs of the Pigeonhole Principle and the Transversal Calculus

Alexander Razborov*
Steklov Mathematical Institute
117966, Moscow, Russia

Avi Wigderson†
Institute of Computer Science
Hebrew University
Jerusalem, Israel

Andrew Yao‡
Computer Science Department
Princeton University
Princeton, New Jersey 08544

Abstract

We investigate read-once branching programs for the following search problem: given a Boolean $m \times n$ matrix with $m > n$, find either an all-zero row, or two 1's in some column. Our primary motivation is that this models regular resolution proofs of the pigeonhole principle PHP_n^m, and that for $m > n^2$ no lower bounds are known for the length of such proofs. We prove exponential lower bounds (for arbitrarily large m!) if we further restrict this model by requiring the branching program *either*

*Part of the work was done while this author was visiting Special Year on Logic and Algorithms at DIMACS, Princeton. Also supported by Russian Basic Research Foundation grant 96-01-01222.

†Part of this work was done while on sabbatical leave at the Institute for Advanced Study and Princeton University, Princeton. This work was supported by USA-Israel BSF grant 92-00106 and by a Wolfson research award administered by the Israeli Academy of Sciences, as well as a Sloan Foundation grant.

‡This work was supported in part by National Science Foundation and DARPA under grant CCR-9627819, and by USA-Israel BSF grant 92-00106.

to finish one row of queries before asking queries about another row (the *row model*) *or* put the dual column restriction (the *column model*).

Then we investigate a special class of resolution proofs for PHP_n^m that operate with positive clauses of rectangular shape; we call this fragment the *rectangular calculus*. We show that all known *upper* bounds on the size of resolution proofs of PHP_n^m actually give rise to proofs in this calculus and, inspired by this fact, also give a remarkably simple "rectangular" reformulation of the Haken-Buss-Turán lower bound for the case $m \ll n^2$. Finally we show that the rectangular calculus is equivalent to the column model on the one hand, and to *transversal calculus* on the other hand, where the latter is a natural proof system for estimating from below the transversal size of set families. In particular, our exponential lower bound for the column model translates both to the rectangular and transversal calculi.

0. Warm-up

The following elementary "data structure" problems, which may be contemplated as independent puzzles by the reader, are the axis connecting the different notions in the title of the paper. Consider algorithms which probe, once, the entries of an input array A in an arbitrary adaptive order, and use s bits of memory. Let $m > n$. What is the smallest memory size $s = s(n, m)$ needed for solving the following problems?

- When $A \in [m]^n$, find a number in $[m]$ missing from A.
- When $A \in [n]^m$, find two entries A containing the same number from $[n]$.

1. Introduction

Complexity of propositional proofs is rapidly taking on as important a role in the theory of feasible proofs as the role played by the complexity of Boolean circuits in the theory of efficient computations. The resolution proof system introduced in [Bla37] and further developed in [DP60, Rob65] is one of the first and simplest in the hierarchy of propositional proof systems; it is also of importance for various automatic theorem proving procedures. Tseitin [Tse68] proved, almost 30 years ago, the first exponential lower bound for regular resolutions (these are resolutions with the additional restriction that along every path every particular variable can be resolved at most once).

However, despite its apparent (and deluding) simplicity, the first lower bounds for *non-regular* resolutions were proven only in 1985 by Haken [Hak85]. These bounds were achieved

for the pigeonhole principle PHP_n^{n+1} which asserts that $n+1$ pigeons cannot sit in n holes so that every pigeon is alone in its hole. Motivated by a separation problem in Bounded Arithmetic (just like the research on the complexity of Boolean circuits is motivated by the needs of the theory of Turing computations!), Buss and Turán [BT88] extended his bound to $\exp\left(\Omega\left(\frac{n^2}{m}\right)\right)$ for a more general form PHP_n^m of the pigeonhole principle in which the number of pigeons, m, is also a parameter. See also [Urq87, CS88, BP96] for other bounds on the complexity of resolutions, and [Juk97] for a generalization of the Haken-Buss-Turán bound to the case of semantic resolutions.

All these lower bounds trivialize when $m \geq n^2$. As mentioned in [BT88] (also see [Kra94, page 31]) it is an open question whether $PHP_n^{n^2}$ has a poly-size resolution proof, and this is open even for regular resolutions. More generally, it is open whether there is any m (as a function of n) for which PHP_n^m has a resolution proof of size polynomial in n. The only non-trivial upper bound is due to Buss and Pitassi [BuP96]: PHP_n^m has a resolution proof of size $\exp(O(\sqrt{n \log n} + n \log n / \log m))$, and the only known lower bound was also proven in the same paper: every *tree-like* resolution proof of PHP_n^m must have size at least 2^n.

In this paper we make some partial progress toward resolving the above open question in the negative for regular (and, perhaps, for general) resolutions. The meaning of our results is most clear when we employ the characterization of regular resolutions in computational terms (see e.g. [Kra94, Theorem 4.2.3]). Namely, regular resolutions are known to be equivalent to read-once branching programs (b.p.) solving the following search problem: given a truth assignment, find some initial clause falsified by this assignment.

For the special case when the search problem (i.e. the initial clauses) corresponds to PHP_n^m, we introduce two restricted classes of read-once b.p. and call these computational models the *row model* and the *column model*. In the row model (with rows corresponding to pigeons) the (read-once) b.p. must query all variables from some row immediately after it queries the first such variable. The column model is defined dually.

We prove a tight $\exp(\Omega(n \log n))$ size lower bound in the row model and an almost tight bound $\exp(\Omega(\sqrt{n} + n/\log m))$ in the column model; note that they make perfect sense for $m = \infty$. The proofs for both models have one remarkable feature in common that is somewhat novel for research of this kind (in fact, it is this feature that allowed us to overcome the n^2 barrier). As in many similar proofs we do construct a distribution on inputs that fools branching programs from some class. But (and this is the novelty) *our distributions depend essentially on the program being fooled, and are being constructed along with the progress of computation itself.*

Both the obvious $2^{O(n)}$-sized resolution proof of PHP_n^m and the Buss-Pitassi proof

mentioned above can be restructured to operate with positive clauses of rectangular form only. Inspired by this fact, we introduce the corresponding *rectangular calculus* (a subclass of resolution proofs for PHP_n^m), and show that both these proofs can be carried out there. We have failed to simulate arbitrary resolution proofs of PHP_n^m in this calculus, and we doubt that such a simulation is possible (it actually seems that already the row model cannot be reduced to the rectangular calculus). However, as will be seen, we have succeeded in "simulating" at least the proof of the Haken-Buss-Turán lower bound: the "rectangular" version of their proof is remarkably simple and may be of independent interest.

Interestingly, it turns out that the rectangular calculus is equivalent to the column model. This allows us to translate the Buss-Pitassi upper bound $\exp(O(\sqrt{n \log n} + n \log n / \log m))$ to the column model, and our $\exp(\Omega(\sqrt{n} + n/\log m))$ lower bound in the column model to the rectangular calculus; these bounds are matching up to a logarithmic factor.

From the complexity-theoretic perspective, the set of all propositional tautologies TAUT is just one of natural *co*-**NP**-complete sets, even though it was historically the first. For any such set we can raise the question of what are natural (i.e., coherent to the intrinsic structure of the set) proof systems for membership proving, and then ask how proof systems for different systems compare to each other in terms of their strength via natural reductions. While natural proof systems for the *co*-**NP**-complete sets corresponding to the INDEPENDENT SET and CHROMATIC NUMBER problems were defined and studied by Chvatal [Ch77] and McDiarmid [Mc79] respectively, no relations between the power of these systems and others are known. Reductions were systematically studied only for the different systems for TAUT (which, in our opinion, is to a large extent caused by historical and psychological reasons). The only nice exceptions we are aware of are Hajós calculus for the set of non-3-colourable graphs [Haj61], and the proof of its equivalence to Frege systems [PU92].

We contribute to this line of research by defining a natural (sound and complete) proof system for proving lower bounds on the transversal (hitting set) size of set families (dual to the SET COVER problem). We call this system the *transversal calculus* and show it to be equivalent to the rectangular calculus. In particular, all upper and lower bounds for the column model and for the rectangular calculus immediately translate to the transversal calculus.

The paper is organized as follows. In Section 2 we recall some needed notation and definitions. In Section 3 we present our lower bounds for the row and column models. Section 4 is devoted to the rectangular calculus (including our reformulation of the Haken-Buss-Turán bound), and Section 5 to the transversal calculus. The paper is concluded

with a few remarks and open problems in Section 6.

2. Preliminaries

Let $p_1, p_2, \ldots, p_n, \ldots$ be propositional *atoms*. A *literal* is either an atom or the negation $(\neg p)$ of an atom p. A *clause* is a set of literals, to be thought of as the disjunction of participating literals. A clause is *positive* if it does not contain negated literals. The *resolution system* is the propositional proof system that operates with clauses and has one rule of inference

$$\frac{C_0 \cup \{p\} \qquad C_1 \cup \{\neg p\}}{C} \ (C_0 \cup C_1 \subseteq C)$$

called the *resolution rule*. We say that the atom p is *resolved* in this application of the resolution rule. A *resolution proof* is a proof in the resolution system. A *resolution refutation* of a set of clauses is a resolution proof of the empty clause from this set.

Throughout this paper we allow straight-line proofs (as opposed to tree-like), i.e., after a formula is inferred, it can be used arbitrarily many times in further inferences. The *size* of a proof is the number of clauses in it.

A resolution proof is *regular* if along every path from an axiom to the final clause every atom is resolved at most once. For any unsatisfiable set of clauses

$$\mathcal{C} = \{C_1(p_1, \ldots, p_n), \ldots, C_k(p_1, \ldots, p_n)\},$$

let us consider the following search problem $S_{\mathcal{C}}$: given a truth assignment $a \in \{0,1\}^n$, find some ν such that $C_\nu(a) = 0$.

Following Borodin and Cook [BC82], we define an *R-way branching program in n variables* as a directed acyclic graph with one source node s (sometimes also called the *root*), in which every non-sink node v is labeled by one of the variables $x_1, \ldots, x_n$ (denoted $var(v)$) and has exactly R outgoing edges numbered by $1, \ldots, R$. Let $[R] = \{1, 2, \ldots, R\}$. Every input string $A \in [R]^n$ determines a *computational path* $comp(A)$ from s to a sink node. An R-way b.p. (branching program) *solves* some search problem with inputs from $[R]^n$ if its sinks can be labeled by possible solutions to the search in such a way that for every $A \in [R]^n$, $comp(A)$ leads to a sink labeled by a solution admissible for the input A.

The *size* of an R-way b.p. is the total number of nodes. The logarithm of size corresponds to the space used by general sequential algorithms.

An R-way b.p. is *read-once* if along every path p every variable is tested (i.e. appears as a node label) at most once. Let $X(p)$ be the set of variables that are tested along a path p. A read-once b.p. is *uniform* if:

a) for a path p beginning at the root s, $X(p)$ depends only on the terminal node v of p (accordingly, we denote it by $X(v)$);

b) for every sink t, $X(t)$ contains all variables.

If, moreover, the variables are tested in the same order along every path (i.e., $X(v)$ depends only on the *depth* of v defined as $|X(v)|$), the program is called *oblivious*.

Uniform read-once R-way b.p. possess the nice property that *every* path from the root to a sink node is the computational path $comp(A)$ for a uniquely defined input $A \in [R]^n$; therefore we can identify inputs with such I/O paths. On the other hand, [Oko91] noticed that uniformity is not actually a serious restriction:

Proposition 2.1. *Every R-way read-once b.p. in n variables can be simulated by an equivalent uniform program whose size is larger by at most a factor of n.*

(It is not known whether a similar simulation is in general possible by oblivious programs.) For completeness we include here a sketch of its proof.

Proof of Proposition 2.1, sketch. Given an *arbitrary* read-once R-way b.p. B, for any non-sink node v, we still define $X(v)$ as the *union* of $X(p)$ taken over all the paths p leading from s to v. For each sink v, define $X(v)$ to be the set of all the variables. If $e = (v, v')$ is an edge of B then clearly $X(v) \cup \{var(v)\} \subseteq X(v')$. Whenever the difference $X(v') \setminus (X(v) \cup \{var(v)\}) = \{x_{i_1}, \ldots, x_{i_d}\}$ is non-empty, we insert d new nodes $v_{e,1}, \ldots, v_{e,d}$ between v and v' and replace e with the following construction. $v_{e,1}$ will be the terminal node of e. $var(v_{e,\nu}) = x_{i_\nu}$, and all R edges going out of $v_{e,\nu}$ go to the same node $v_{e,\nu+1}$ (to v' if $\nu = d$).

It is easy to see that this procedure transforms B into a uniform program computing the same function and increases the number of nodes in B by at most a factor of n.■

The following remarkable result is apparently the only known case of an equivalence between a propositional proof system and a computational model (we omit the prefix "2-way" in the case of ordinary binary programs):

Proposition 2.2. *Let $\mathcal{C}$ be an unsatisfiable set of clauses. Then the minimum size of any regular resolution refutation of $\mathcal{C}$ is equal to the minimum size of any read-once b.p. solving $S_{\mathcal{C}}$.*

The proof can be found e.g. in [Kra94, Theorem 4.3]. For completeness we include here its sketch as well.

Proof of Proposition 2.2, sketch. There are two directions to prove. In both, the transformation of one model to the other leaves the underlying directed graph untouched (so in particular size is unchanged), and simply replaces the labels.

First, assume we are given a read-once b.p. B solving $S_{\mathcal{C}}$. Let v be an internal (non-sink) vertex in B, and v_0 (resp. v_1) the descendant of v reached if $var(v) = 0$ (resp. $var(v) = 1$).

Inductively label the vertices of B by clauses $cl(v)$ for every v as follows. If v is a sink, then $cl(v)$ is its label in B. If some vertex is unlabeled, consider any such vertex v for which v_0 and v_1 are already labeled. Let $var(v) = p$. If $cl(v_0)$ is NOT of the form $C_0 \cup \{p\}$ (for some C_0), set $cl(v) \leftarrow cl(v_0)$. If $cl(v_1)$ is NOT of the form $C_1 \cup \{\neg p\}$ (for some C_1), set $cl(v) \leftarrow cl(v_1)$. Otherwise, set $cl(v) \leftarrow C_0 \cup C_1$.

When all vertices are labeled, reversing the direction of edges in this dag gives a resolution refutation of $S_{\mathcal{C}}$. This is proved by simple induction showing that if after each such step we remove from B the edges (v, v_0) and (v, v_1) (thus creating a new sink v), then the resulting b.p. is a read-once b.p. solving $S_{\mathcal{C} \cup \{cl(v)\}}$. (Note that the fact B is read-once is essential for the correctness of the induction step!)

For the second (easy) direction, assume we are given a regular resolution refutation R of $\mathcal{C}$. Construct the b.p. as follows. For each clause C in $\mathcal{C}$ make a separate vertex v_C. For every step of the proof do the following. Assume this step derives a new clause C from $C_0 \cup \{p\}$ and $C_1 \cup \{\neg p\}$; let $v_0 = v_{C_0 \cup \{p\}}$ and $v_1 = v_{C_1 \cup \{\neg p\}}$. Label v_C by the variable p, and connect it with edges labeled 0 (resp. 1) to v_0 (resp. v_1).

Again, simple induction shows that starting from a newly added vertex v_C, and following the computational path according to any truth assignment a which falsifies C, we eventually reach a sink $v_{C'}$ such that C' is also falsified by a. Since the last vertex generated in the construction above is $v_\emptyset$, and by virtue of the regularity of R, it yields a read-once b.p. for $S_{\mathcal{C}}$.■

Definition 2.3. $\neg PHP_n^m$ is the following set of clauses, over the ($m \times n$ matrix of) atoms p_{ij}

$$\{\neg p_{i_1,j}, \neg p_{i_2,j}\} \ (i_1, i_2 \in [m],\ i_1 \neq i_2;\ j \in [n]); \tag{1}$$

$$\{p_{i1}, p_{i2}, \ldots, p_{in}\} \ (i \in [m]). \tag{2}$$

Clearly, $\neg PHP_n^m$ is unsatisfiable for $m \geq n+1$. Hence it possesses resolution refutations that we will sometimes call *resolution proofs of* PHP_n^m. In the matrix representation[1], an

[1]Note that this is a transpose of Haken's original representation from [Hak85]. The reason for implementing this change is that at the moment it has become more customary to use the notation in which the first index i corresponds to the largest of the two numbers m, n (most often m).

admissible solution of $S_{\neg PHP_n^m}$ is either an identically zero row, or two 1-entries in the same column. Since we do not consider in this paper any tautologies other than PHP_n^m, we assume throughout that m, n are some integers, $m \geq n+1$, and all propositional atoms have the form p_{ij}, where $i \in [m]$ and $j \in [n]$.

Let $\mathcal{A}$ be a family of sets (that are subsets of some finite underlying universe). A set T is called a *transversal* of $\mathcal{A}$ if it intersects all members of $\mathcal{A}$ (i.e., $A \cap T \neq \emptyset$ for all $A \in \mathcal{A}$). The *transversal number* $\tau(\mathcal{A})$ of the family $\mathcal{A}$ is the minimum size $|T|$ of any transversal T of $\mathcal{A}$.

3. Lower bounds

The proofs of many lower bounds in Boolean complexity are based upon the following transparent idea: define a natural probability distribution $\boldsymbol{a}$ on inputs[2], and show that every small circuit/program B presumably solving our problem must err with positive probability on a random input chosen accordingly to $\boldsymbol{a}$. In particular, it seems that all known lower bounds for read-once b.p. (see e.g. [Weg87, Chapter 14.4], [Raz91] for examples) employ this idea.

In this paper we bring something fresh to this method: the distribution $\boldsymbol{a}_B$ will *not* be fixed in advance but will depend essentially on the program B, and will be constructed dynamically along with the progress of the computation. We consider two types of read-once b.p. for $S_{\neg PHP_n^m}$: those which must query all variables from some row immediately after querying the first such variable, and those satisfying the dual column restriction.

3.1. The Row Model

If a read-once b.p. attempting to solve $S_{\neg PHP_n^m}$ queries at once all variables from some row, then the adversary should not respond with all zeros since then the program can immediately produce an unsatisfiable clause of the form (2). Conversely, if he follows this recommendation and never responds with all zeros, then all clauses (2) will be satisfied, and the result of the search must be a negative clause of the form (1). Which means that it is disadvantageous for the adversary to respond with more than a single one either, and this leads us to the following model:

[2]throughout the paper we use the math bold face for denoting random objects

Definition 3.1. In the *row model*, an n-way read-once b.p. in m variables attempts to output a solution to the following search problem Row_n^m: given an input $A \in [n]^m$, find some $i_1, i_2 \in [m]$ and $j \in [n]$, where $i_1 \neq i_2$, such that $A_{i_1} = A_{i_2} = j$.

Clearly, there is an $\exp(O(n \log n))$-sized program solving Row_n^m: just ignore all but the first $(n+1)$ variables, and treat every one of n^{n+1} inputs individually by a decision tree. Our first result shows that in the row model we cannot do any better:

Theorem 3.2. *Any n-way read-once b.p. in m variables that solves Row_n^m must have size $\exp(\Omega(n \log n))$.*

Proof. We can assume $n \geq 3$. Let B be an n-way read-once b.p. in m variables. For any node v, denote by $J(v)$ the set of all $j \in [n]$ such that for some *fixed* variable x_i, *every* path from the root s to v makes the assignment $x_i = j$. Note that if $e = (u, v)$ is an edge (directed from u to v), then $|J(v)| \leq |J(u)| + 1$. Let us call an edge e labeled by j and outgoing of v *legal* if $j \notin J(v)$ and *illegal* otherwise.

Claim 3.3. *If B solves Row_n^m, then there is no path from the root to a sink consisting entirely of legal edges.*

Proof of Claim 3.3. Consider some path p between the root s and a sink node t labeled by $A_{i_1} = A_{i_2} = j$. Then p must contain at least two edges labeled by j. Let $e = (v, u)$ be the last edge along p with this property. We are going to show that e is illegal.

Replacing, if necessary, i_1 by i_2, we may assume that i_1 is *not* the label of v. Every path from s to v must make the assignment $x_{i_1} = j$: otherwise we could combine it with the segment of p beginning at v (keeping in mind that B is a read-once program), and get a computational $s - t$ path that does not make the assignment $x_{i_1} = j$, contrary to the assumption that B solves Row_n^m. Hence $j \in J(v)$, and e is illegal.■

Now we convert B into a finite Markov chain as follows: the set of states is simply the set of nodes, s is the initial state, and terminal states are sink nodes along with those v for which $J(v) = [n]$. The Markov process which at any non-terminal state v traverses all outgoing *legal* edges with equal probabilities, defines a random path $\boldsymbol{p}_B$. Claim 3.3 implies that with probability 1 $\boldsymbol{p}_B$ actually arrives at a terminal node v of the second type, i.e. such that $J(v) = [n]$. Also, the value $|J(v)|$ can increase, decrease, or stay the same along each edge in $\boldsymbol{p}_B$. But every time it increases, it increases by at most one in a step. Thus, with probability 1, $\boldsymbol{p}_B$ visits some node v such that $|J(v)| = \lceil n/2 \rceil$. Let $\boldsymbol{v}$ be the *first* such

node along $\boldsymbol{p}_B$. We are only left to show that for every specific v_0 with $|J(v_0)| = \lceil n/2 \rceil$, $\mathbf{P}[\boldsymbol{v} = v_0] \leq \exp(-\Omega(n \log n))$ (and, hence, there must be at least $\exp(\Omega(n \log n))$ such v_0).

Consider any v_0 with $J(v_0) = \{j_1, \ldots, j_{\lceil n/2 \rceil}\}$, and let i_ν ($1 \leq \nu \leq \lceil n/2 \rceil$) be integers such that every path from s to v_0 has made all the assignments $x_{i_1} = j_1, \ldots, x_{i_{\lceil n/2 \rceil}} = j_{\lceil n/2 \rceil}$. Clearly, $i_1, \ldots, i_{\lceil n/2 \rceil}$ are also distinct. Then $\boldsymbol{v} = v_0$ implies, in particular, that before arriving at the node v_0, the Markov process $\boldsymbol{p}_B$ must have tested all variables $x_{i_1}, \ldots, x_{i_{\lceil n/2 \rceil}}$ (possibly in a variable order) and make every time the decision $x_{i_\nu} = j_\nu$. Moreover, since $\boldsymbol{v}$ was chosen to be the *first* node along $\boldsymbol{p}_B$ with $J(v) = \lceil n/2 \rceil$, $\boldsymbol{p}_B$ must make these decisions at nodes v with at least $\lceil n/2 \rceil$ outgoing legal edges which implies that, for each ν, the probability to make the decision $x_{i_\nu} = j_\nu$ is at most $2/n$. It follows from general properties of Markov processes that $\mathbf{P}[\boldsymbol{v} = v_0] \leq (2/n)^{\lceil n/2 \rceil} \leq \exp(-\Omega(n \log n))$.

The proof of Theorem 3.2 is complete.■

3.2. The Column Model

Similarly to the row model, if a read-once b.p. for $S_{\neg PHP_n^m}$ always queries at once all variables from the same column, we may assume that it receives in response a single one, and this leads us to the following model that is dual to the row model:

Definition 3.4. Let Column_n^m be the following search problem: given an input $A \in [m]^n$ (viewed as a function), find some $i \notin \mathrm{im}(A)$. In the *column model*, we consider m-way read-once b.p. in n variables attempting to solve Column_n^m.

Unlike the row model, there *is* a non-trivial upper bound in this model, and it will be presented in the next section (see Corollary 4.5). Our lower bound matches it within a factor of $O(\log n)$ in the exponent:

Theorem 3.5. *Any m-way read-once b.p. in n variables that solves Column_n^m must have size at least $\exp(\Omega(\sqrt{n} + n/\log m))$.*

Proof. First we prove the bound

$$\exp(\Omega(n/\log m)). \tag{3}$$

Let B be an m-way read-once b.p. in n variables solving Column_n^m. By Proposition 2.1, we may assume that B is uniform. For a node v of B denote by $I(v)$ the set of all $i \in [m]$ which are *not* assigned to any variable $x_j \in X(v)$ along any path from the root s to v. Let us call an edge e outgoing of v and labeled by i *legal* if $i \in I(v)$ and *illegal* otherwise.

The dual statement to Claim 3.3 simply says that $I(v) \neq \emptyset$ for every node v. Moreover, $I(s) = [m]$, $I(v)$ can only decrease along edges, and $i \in I(t)$ for every sink node t labeled by $i \notin \mathrm{im}(A)$. Define $\boldsymbol{p}_B$ by the same Markov process as in the proof of Theorem 3.2 (with the new notion of legal edge, of course). The remark above implies that $\boldsymbol{p}_B$ arrives, with probability 1, to a sink node $\boldsymbol{t}$. Since B is uniform, $\boldsymbol{p}_B$ has length n (w.p. 1). Let $k = \lceil \log m \rceil$, and $s = v_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_k = \boldsymbol{t}$ be nodes along $\boldsymbol{p}_B$ that divide this random path into segments of length at least $\lfloor n/k \rfloor$ each.

Since $|I(v_0)| = m$, $|I(\boldsymbol{v}_k)| \geq 1$ and $I(\boldsymbol{v}_\nu)$ is decreasing in ν (w.p. 1), we have that for some $0 \leq \nu \leq k-1$,

$$|I(\boldsymbol{v}_{\nu+1})| \geq \frac{1}{2}|I(\boldsymbol{v}_\nu)|.$$

Similarly to the proof of Theorem 3.2, we are left to show that for any specific pair (u_0, u_1) of nodes with the properties $|X(u_1) \setminus X(u_0)| \geq \lfloor n/k \rfloor$, $I(u_1) \subseteq I(u_0)$ and $|I(u_1)| \geq \frac{1}{2}|I(u_0)|$, we have

$$\mathbf{P}[u_0 \text{ and } u_1 \text{ belong to } \boldsymbol{p}_B \text{ in this order}] \leq 2^{-\lfloor n/k \rfloor}. \tag{4}$$

This again follows from the general theory of Markov processes. Indeed, any successful $\boldsymbol{p}_B$ can visit between u_0 and u_1 only those nodes v for which

$$I(u_1) \subseteq I(v) \subseteq I(u_0). \tag{5}$$

At any such node v there are $|I(v)|$ outgoing legal edges, and at most $|I(v) \setminus I(u_1)|$ ways for the Markov process to maintain the property (5). Thus, the probability to make the "right" decision at every individual node v is at most

$$\frac{|I(v)| - |I(u_1)|}{|I(v)|} \leq 1 - \frac{|I(u_1)|}{|I(u_0)|} \leq \frac{1}{2},$$

and on its way from u_0 to u_1 the process must make at least $\lfloor n/k \rfloor$ of them. The bounds (4) and (3) follow.

In order to see the remaining bound $\exp(\Omega(\sqrt{n}))$ on the size of B, let m' be the overall number of sinks in B. If $m' \geq 2^{\sqrt{n}}$, we are done. If $m' \leq 2^{\sqrt{n}}$, we can assume w.l.o.g. that only $1, 2, \ldots, m'$ appear as labels on the sink nodes in B. It is easy to see that, without increasing its size, B can be transformed into an m'-way read-once b.p. B' solving $\mathrm{Column}_n^{m'}$. Now the bound $\exp(\Omega(\sqrt{n}))$ follows from the already proven (3) when substituting $m' \leq 2^{\sqrt{n}}$ for m.

The proof of Theorem 3.5 is complete.■

4. The Rectangular Calculus

For $I \subseteq [m],\ J \subseteq [n]$, let R_{IJ} denote the positive clause $\{p_{ij} \mid i \in I,\ j \in J\}$; we call clauses of this form *rectangular* or, if we want to specify sizes, $|I| \times |J|$ *rectangular*. The *perimeter* of a non-empty $a \times b$ rectangular clause is defined as $a + b$ (half of the "geometrical" perimeter).

As we will see below, rectangular clauses (and especially those of perimeter $(n+1)$) are of extreme importance for both upper and lower bounds on the complexity of resolution proofs of PHP_n^m. This motivates the study of the following fragment of resolutions that operates with rectangular clauses only and captures that kind of reasoning.

Definition 4.1. The *rectangular calculus* is the proof system that works with rectangular clauses and has one inference rule

$$\left.\begin{array}{c} \dfrac{R_{I_1,J_1\cup\{j\}},\ldots,R_{I_k,J_k\cup\{j\}}}{R_{IJ}} \\ (I_1\cap\ldots\cap I_k=\emptyset,\ j\in[n],\ I_1\cup\ldots\cup I_k\subseteq I,\ J_1\cup\ldots\cup J_k\subseteq J). \end{array}\right\} \quad (6)$$

(Intuitively, only one pigeon from I can go to hole j, and there is no pigeon which is common to all I_j, so at least one has to go to J.)

A *rectangular proof* is a proof in the rectangular calculus, a *rectangular refutation* of a set of rectangular clauses is a rectangular proof of the empty clause from this set, and a *rectangular refutation of* $\neg PHP_n^m$ (a *rectangular proof of* PHP_n^m) is a rectangular refutation of the set of axioms (2). The *size* of a rectangular proof is the number of clauses in it. Let $s(m,n)$ be the minimum size of any rectangular refutation of $\neg PHP_n^m$.

Let us firstly see that proofs in the rectangular calculus can be polynomially simulated by resolution proofs from $\neg PHP_n^m$.

Statement 4.2. *Suppose that a rectangular clause R has a rectangular proof of size s from a set $\mathcal{R}$ of initial rectangular clauses. Then there exists a resolution proof of R from the set of axioms $\mathcal{R} + (1)$ that has size at most $m^2(s+n)$.*

Proof. Since there are at most $m^2 n$ axioms (1), we only have to show how to simulate the rule (6) with at most m^2 resolution inferences using (1) as additional axioms. This is done

straightforwardly: for every $i \in I_1$ we find some ν with $i \notin I_\nu$, and infer $R_{IJ} \cup \{\neg p_{ij}\}$ from $R_{I_\nu, J_\nu \cup \{j\}}$ using at most $|I_\nu| \leq (m-1)$ resolutions with appropriate axioms (1). Then we consecutively resolve the resulted clauses with $R_{I_1, J_1 \cup \{j\}}$ along $\{p_{ij} \mid i \in I_1\}$ and get rid of these atoms. The whole inference uses at most $|I_1| \cdot (m-1) + |I_1| \leq m^2$ resolution rules.■

Unfortunately, it does not look plausible that arbitrary resolution proofs of PHP_n^m can be efficiently simulated in the rectangular calculus. However, as the following examples show, many known *constructions*, both in the context of upper and lower bounds, can in fact be viewed as rectangular.

Example 1 (brute-force search proof of PHP_n^{n+1}). We consecutively infer, for $d = n, n-1, \ldots, 1, 0$, all $(n+1-d) \times d$ rectangular clauses of the form $R_{I,[d]}$. The case $d = n$ is given as axioms (2), and if we already have all $(n-d) \times (d+1)$ clauses, then, given $I \subseteq [n+1]$ with $|I| = n-d+1$, we infer $R_{I,[d]}$ from $\left\{R_{I\setminus\{i\},[d+1]} \mid i \in I\right\}$ with a single application of (6). At the end (for $d = 0$) we get the empty rectangle. This shows the bound

$$s(n+1, n) \leq 2^{n+1}. \tag{7}$$

Example 2 (non-trivial proof of PHP_n^m for large m [BuP96]). We also have the following non-trivial recursion for $s(m, n)$:

$$s(m^2, 2n) \leq (m+1) \cdot s(m, n). \tag{8}$$

In order to see this, let P be the rectangular refutation of $\neg PHP_n^m$ that has size $s(m, n)$. Replacing every clause R_{IJ} in P with $R_{I, J \cup \{n+1, \ldots, 2n\}}$ (i.e., adding n new holes), we will get a rectangular proof of $R_{[m], \{n+1, \ldots, 2n\}}$ from axioms $\{p_{i1}, p_{i2}, \ldots, p_{i,2n}\}$ $(i \in [m])$ that has the same size $s(m, n)$. By symmetry, we have similar $s(m, n)$-sized proofs of every $m \times n$ rectangular clause.

The dual transformation that can be done with the proof P is to replace every single pigeon with an m-member pigeon family, i.e., replace every clause R_{IJ} with $R_{I \times [m], J}$. This gives us an $s(m, n)$-sized rectangular refutation of the set of clauses

$$\left\{R_{\{1\} \times [m], \{n+1, \ldots, 2n\}}, \ldots, R_{\{m\} \times [m], \{n+1, \ldots, 2n\}}\right\}.$$

Combining this refutation with the $s(m, n)$-sized proofs of $R_{\{i\} \times [m], \{n+1, \ldots, 2n\}}$ constructed above, we get the recursion (8).

Now, (7) and (8) imply

$$s\left((n+1)^{(2^\ell)}, 2^\ell \cdot n\right) \leq (n+1)^{(2^\ell)} \cdot 2^{n+1}.$$

Substituting here $n := \lceil \frac{2n \log n}{\log m} \rceil$, $\ell := \lfloor \log\left(\frac{\log m}{\log n}\right) \rfloor$ ($m \geq n^3$, logarithms are base 2), we have the bound

$$s(m,n) \leq \exp(O(\log m + n \log n / \log m)) \tag{9}$$

(which follows from (7) for $m < n^3$). It implies

$$s(m,n) \leq \exp\left(O\left(\sqrt{n \log n} + n \log n / \log m\right)\right) \tag{10}$$

(for $m \geq 2^{\sqrt{n \log n}}$ just use the first $2^{\sqrt{n \log n}}$ pigeons ignoring all others), and this is the best upper bound on the complexity of resolution proofs of PHP_n^m (not necessarily rectangular!) known today.

Example 3 (Haken-Buss-Turán bound). In Example 1 we used rectangular clauses of perimeter $(n+1)$ for upper bounds; now we show how to trace such clauses through an *arbitrary* resolution proof.

Firstly we get rid of negations (a dual construction was previously used in [Bus87, BP96, BuP96, Juk97]). For this purpose we replace in a resolution refutation of $\neg PHP_n^m$ every negative literal $(\neg p_{ij})$ by the set of literals $\{p_{ij'} \mid j' \neq j\}$. This results in a proof of the empty clause in the *positive calculus* that operates with positive clauses, has $1 \times n$ and $2 \times (n-1)$ rectangular clauses as axioms (the latter resulting from (1)), and has one inference rule

$$\frac{C_1 \cup \{p_{ij}\} \qquad C_2 \cup \{p_{ij'} \mid j' \neq j\}}{C} \ (C_1 \cup C_2 \subseteq C). \tag{11}$$

In fact, this calculus is easily seen to be equivalent to resolution proofs from $\neg PHP_n^m + \{\{\neg p_{ij_1}, \neg p_{ij_2}\} \mid i \in [m];\ j_1, j_2 \in [n],\ j_1 \neq j_2\}$, but we will not need this in what follows.

Now suppose that the premises in (11) are known to contain rectangular clauses of perimeter $(n+1)$: $R_{I_1 J_1} \subseteq C_1 \cup \{p_{ij}\}$; $R_{I_2 J_2} \subseteq C_2 \cup \{p_{ij'} \mid j' \neq j\}$. We wish to find a subclause of perimeter $(n+1)$ in the conclusion C. We may assume that $i \in I_1 \cap I_2$ and $j \in J_1 \setminus J_2$ (otherwise C simply inherits one of $R_{I_1 J_1}, R_{I_2 J_2}$). But then C contains two rectangular clauses $R_{(I_1 \cap I_2) \setminus \{i\}, J_1 \cup J_2}$ and $R_{I_1 \cup I_2, J_1 \cap J_2}$, and the sum of their perimeters is equal to $|I_1 \cap I_2| + |J_1 \cup J_2| + |I_1 \cup I_2| + |J_1 \cap J_2| - 1 = |I_1| + |I_2| + |J_1| + |J_2| - 1 = 2n+1$. Hence, one of them has perimeter at least $(n+1)$.

Summing up, in every line of a refutation in the positive calculus we can trace a rectangular clause of perimeter $(n+1)$, until we get at the end the "virtual" $(n+1) \times 0$ empty clause. Moreover, the above construction shows that $|I| \leq |I_1| + |I_2|$, where I_1, I_2, I correspond to the rectangular clauses in the premises and the conclusion of the rule (11), respectively. Since initially for every axiom we have $|I| \leq 2$, every such refutation should

contain somewhere an $(n/3) \times (n/3)$ rectangular subclause ("bottleneck" in the standard terminology).

Now it is easy to finish the proof of the Haken-Buss-Turán $\exp(\Omega(n^2/m))$ bound with the idea proposed in [BP96]. Namely, if we hit a refutation in the positive calculus with a random restriction assigning $n/2$ randomly chosen pigeons to $n/2$ randomly chosen holes, then every individual clause containing an $(n/6) \times (n/6)$ rectangular subclause gets killed to 1 with probability at least $1 - \exp(-\Omega(n^2/m))$. Hence, in order for the restricted refutation to have an $(n/6) \times (n/6)$ bottleneck, the original refutation must contain at least $\exp(\Omega(n^2/m))$ positive clauses.

Quite remarkably, the rectangular calculus is equivalent to the column model from the previous section. The proof of this equivalence (similar to the proof of Proposition 2.2) takes up the rest of Section 4.

Firstly we note that rectangular proofs can be further structured to work with only "one-dimensional" clauses like those used in Example 1. We say that a rectangular clause R_{IJ} is *compact* if $J = [d]$ for some d, and abbreviate this clause as $R_{I,d}$. Note that the axioms (2) are compact. The following inference rule

$$\left.\begin{array}{c} \dfrac{R_{I_1,d+1}, \ldots, R_{I_k,d+1}}{R_{I,d}} \\ (I_1 \cap \ldots \cap I_k = \emptyset,\ d \geq 0,\ I_1 \cup \ldots \cup I_k \subseteq I) \end{array}\right\} \quad (12)$$

is a special case of the rule (6).

Lemma 4.3. *For every rectangular proof of a compact clause R from a set $\{R_1, \ldots, R_k\}$ of initial compact clauses, there exists a rectangular proof of R from $\{R_1, \ldots, R_k\}$ whose size is less or equal to the size of the original proof, and such that every line is a compact clause and every inference rule has the form* (12).

Proof. It is easy to see that the "compression operator" that replaces any rectangular clause R_{IJ} by $R_{I,|J|}$, transforms the rule (6) into either the trivial weakening rule (i.e., when some premise is contained in the conclusion) or an instance of (12). Now we contract all applications of weakening rules using the fact that the restriction put on the conclusion $R_{I,d}$ in (12) is monotone in both I and d.■

Theorem 4.4. *$s(m,n)$ is equal to the minimal possible size of a uniform m-way read-once b.p. in n variables solving the search problem* Column_n^m.

Proof. a). Let P be a rectangular refutation of $\neg PHP_n^m$ that has size $s(m,n)$. By Lemma 4.3 we may assume that P contains only compact clauses, and uses only inference rules of the form (12). We convert P into an (oblivious) m-way read-once b.p. B as follows. Nodes of B are just lines of P, the source node s is the final (empty) rectangle in P, and axioms become sink nodes. For the computational node corresponding to the conclusion $R_{I,d}$ of the inference (12), query $A_{d+1} =?$ is asked, and the outgoing edge labeled by i goes to some premise $R_{I_\nu,d+1}$ with the property $i \notin I_\nu$. Clearly, working on an input $A \in [m]^n$, this b.p. traverses only through compact clauses $R_{I,d}$ falsified by A (in the sense $\{A(1), A(2), \ldots, A(d)\} \cap I = \emptyset$) and thus eventually finds $i \notin \mathrm{im}(A)$.

b). Conversely, suppose that B is a uniform m-way read-once b.p. in n variables solving the search problem Column_n^m. Using notation from the proof of Theorem 3.5, we associate with every node v the compact clause $R(v) = R_{I(v),X(v)}$. Clearly, $R(s)$ is empty (since $X(s)$ is so), $R(t)$ contains $R_{\{i\},[n]}$ for a sink t labeled by the output $i \notin \mathrm{im}(A)$, and $R(v)$ can be obtained from $R(v_1), \ldots, R(v_m)$ via one application of the rule (6) if v is a computational node and $v_1, \ldots, v_m$ are all its children. Thus, we have constructed a rectangular refutation of $\neg PHP_n^m$ that has the same size as B.■

As a by-product of the above proof we obtain the fact that every read-once b.p. solving Column_n^m can be made oblivious without any increase in size (this is also easy to prove directly).

Using Theorem 4.4, we can translate the Buss-Pitassi upper bound from Example 2 to the column model, and our lower bound in Theorem 3.5 to the rectangular calculus:

Corollary 4.5. *There exists an m-way read-once b.p. in n variables that solves* Column_n^m *and has size at most* $\exp(O(\sqrt{n \log n} + n \log n / \log m))$.

Corollary 4.6. $s(m,n) \geq \exp(\Omega(\sqrt{n} + n/\log m))$.

5. The Transversal Calculus

In this section we define a natural (sound and complete) proof system for proving lower bounds on the transversal number $\tau(\mathcal{A})$, and show it to be equivalent to the rectangular calculus.

Recall that for a family of sets $\mathcal{A}$, we denoted by $\tau(\mathcal{A})$ the size of the smallest set hitting every member of $\mathcal{A}$. Let us further define $\cap\mathcal{A} = \cap_{A\in\mathcal{A}}A$ and $\cup\mathcal{A} = \cup_{A\in\mathcal{A}}A$ (respectively the intersection and union of all sets in $\mathcal{A}$). As usual $|\mathcal{A}|$ denotes the cardinality of this family, i.e., the number of sets in $\mathcal{A}$.

Definition 5.1. Lines in the *transversal calculus* have the form $\tau(\mathcal{A}) \geq n$, where $\mathcal{A}$ is a family of sets, and n is an integer. The default axioms are of the form $\tau(\mathcal{A}) \geq 1$, where $\mathcal{A}$ is non-empty, and the only (unary) inference rule has the form

$$\left.\begin{array}{c} \dfrac{\tau(\mathcal{A}) \geq n}{\tau(\mathcal{B}) \geq n+1} \\ (\forall A \in \mathcal{A}\ \exists \mathcal{B}_A \subseteq \mathcal{B}\,(\cap\mathcal{B}_A = \emptyset\ \ \&\ \ \cup\mathcal{B}_A \subseteq A)). \end{array}\right\} \quad (13)$$

While the intuition behind this inference rule may not be clear at first sight, the simple proof of its soundness and completeness below would clarify it. We define the *size* of a transversal proof as the sum of cardinalities $|\mathcal{A}|$ of families appearing in all lines of the proof.

Remark 5.2. A sensible alternative definition of size is to use the count $\sum_{A\in\mathcal{A}}|A|$ in place of $|\mathcal{A}|$ which is tantamount to the length of the proof. These two definitions are polynomially equivalent in many situations, such as for example, if the cardinality of the family we are interested in is not smaller than the number of elements in the underlying universe.

Theorem 5.3. *$\tau(\mathcal{A}) \geq n$ is provable in the transversal calculus if and only if it is true, i.e., this calculus is sound and complete.*

Proof.
Soundness is proved by induction on the length of a transversal proof. For the inductive step, suppose $\tau(\mathcal{A}) \geq n$ is already known to be true, and $\forall A \in \mathcal{A}\ \exists\mathcal{B}_A \subseteq \mathcal{B}(\cap\mathcal{B}_A = \emptyset\ \ \&\ \ \cup\mathcal{B}_A \subseteq A)$. We wish to prove $\tau(\mathcal{B}) \geq n+1$. Assume that, to the contrary, T is a transversal of $\mathcal{B}$ with $|T| = n$. We derive a contradiction. Choose any $i \in T$. Since $T \setminus \{i\}$ is not a transversal of $\mathcal{A}$, there exists some $A \in \mathcal{A}$ such that $A \cap T \subseteq \{i\}$. But since $\cap\mathcal{B}_A = \emptyset$, there exists also some $B \in \mathcal{B}_A$ (which implies $B \subseteq A$) such that $i \notin B$. We conclude that $B \cap T \subseteq (A \cap T) - \{i\} = \emptyset$, contrary to our assumption that T is a transversal of B.

Completeness. Let $\mathcal{U}_n$ be the family of all sets whose complements (to the whole universe) have size $n-1$. Let $\preceq$ be the quasiordering on families of sets given by $\mathcal{A} \preceq \mathcal{B} \Longleftrightarrow \forall B \in \mathcal{B}\ \exists A \in \mathcal{A}(A \subseteq B)$. Completeness is immediately implied by the combination of the following three facts easily checkable individually:

- Provability in the transversal calculus is antimonotone w.r.t. $\preceq$. In other words, if $\mathcal{A} \preceq \mathcal{B}$ and $\tau(\mathcal{B}) \geq n$ is provable, then $\tau(\mathcal{A}) \geq n$ is provable, too.
- $\tau(\mathcal{A}) \geq n$ iff $\mathcal{A} \preceq \mathcal{U}_n$.
- (cf. Example 1) $\tau(\mathcal{U}_n) \geq n$ is provable in the transversal calculus. In fact,

$$\frac{\tau(\mathcal{U}_n) \geq n}{\tau(\mathcal{U}_{n+1}) \geq n+1}$$

  is a legal inference.

■

For a family of sets $\mathcal{A}$ and an integer n such that $\tau(\mathcal{A}) \geq n+1$ is true, let us denote by $t(\mathcal{A}, n)$ the minimum size of any transversal proof of this fact, and by $s(\mathcal{A}, n)$ the minimum size of any rectangular proof of the empty clause from the set of initial clauses

$$\left\{ R_{A,[n]} \mid A \in \mathcal{A} \right\}. \tag{14}$$

Note that $s(\mathcal{A}, n)$ generalizes the function $s(m, n)$ studied in the previous section: namely, $s(m, n) = s(\{\{1\}, \{2\}, \ldots, \{m\}\}, n)$. The following result says that the rectangular and transversal calculi are essentially just different forms of the same proof system:

Theorem 5.4. *For every family of sets $\mathcal{A}$ and every integer n such that $\tau(\mathcal{A}) \geq n+1$, we have*

$$s(\mathcal{A}, n) \leq t(\mathcal{A}, n) \leq s(\mathcal{A}, n) + |\mathcal{A}|.$$

Proof.
Lower bound on $t(\mathcal{A}, n)$. Suppose we have a transversal proof

$$\cfrac{\cfrac{\tau(\mathcal{A}_1) \geq 1}{\tau(\mathcal{A}_2) \geq 2}}{\cfrac{\cdots}{\tau(\mathcal{A}_{n+1}) \geq n+1}}$$

of size $t(\mathcal{A}, n)$, where $\mathcal{A}_{n+1} = \mathcal{A}$. We convert it into a rectangular proof (in the compact form) as follows: for every $A \in \mathcal{A}_d$, introduce the clause $R_{A,[d-1]}$. Then the clauses resulting from $\mathcal{A}_{n+1}$ become initial axioms (14). Furthermore, if $d \leq n$ and $A \in \mathcal{A}_d$, then $R_{A,[d-1]}$ is inferred from $\left\{ R_{B,[d]} \,|\, B \in \mathcal{B}_A \right\}$ (where $\mathcal{B}_A \subseteq \mathcal{A}_{d+1}$ is chosen accordingly to (13)) via one application of (12). Finally, any $A \in \mathcal{A}_1$ (remember that $\mathcal{A}_1$ is non-empty!) gives rise to the empty clause.

Upper bound on $t(\mathcal{A}, n)$. We prove it by reversing the above argument. By Lemma 4.3, there is a rectangular proof in compact form of size $s(\mathcal{A}, n)$. To obtain a transversal proof, we set for each d

$$\mathcal{A}_d = \left\{ A \,\middle|\, R_{A,[d-1]} \text{ appears in the proof} \right\}.$$

One subtle point is that in this way we obtain only a $s(\mathcal{A}, n)$-sized transversal proof of $\tau(\mathcal{A}') \geq n+1$ for some *subset* $\mathcal{A}'$ of $\mathcal{A}$, as we do not require that *all* axioms necessarily appear in the proof (this is more than just an excessive pedantry – cf. the last argument in the proof of the bound (10)!) We convert it into a transversal proof of $\tau(\mathcal{A}) \geq n+1$ simply by adding all sets from $\mathcal{A} \setminus \mathcal{A}'$ to the last line.■

Denote $t(\{\{1\}, \{2\}, \ldots, \{m\}\}, n)$ by $t(m, n)$.

Corollary 5.5. $\exp(\Omega(\log m + n/\log m)) \leq t(m,n) \leq \exp(O(\log m + n \log n/\log m))$.

Proof. Immediate from Theorem 5.4, Corollary 4.6 combined with the trivial bound $t(m,n) \geq m$, and (9).■

6. Conclusion and open problems

In studying the complexity of resolution proofs of the pigeonhole principle PHP_n^m, the case of $m = n^2$ pigeons becomes a natural barrier where ordinary (static) distributions on the set of partial truth assignments, restrictions etc. fail to fulfill their mission. In this paper we have proved first partial results *beyond this barrier*, and we hope that the idea which allowed us to overcome it (i.e., the usage of dynamical distributions constructed along with the progress of a computation or a proof itself) will eventually lead to establishing lower bounds on the size of resolution proofs of PHP_n^m, at least in the regular case. The next step in carrying out this program might be the following

Problem 1. Prove exponential lower bounds on the size of any *oblivious* read-once b.p. solving $S_{\neg PHP_n^m}$.

More modest (but still interesting) goal is to close the logarithmic gap between upper and lower bounds on $s(m,n)$:

Problem 2. What is the order of magnitude of $\log s(\infty,n)$? [BuP96] showed that it is at most $\sqrt{n\log n}$, and we have proved that it is at least $\sqrt{n}$.

Finally, we would like once more to draw attention to the fact that we have only a handful of natural proof systems for *co*-**NP**-complete sets other than TAUT. We propose a more systematic study of natural reducibilities between such systems: this would help convincing combinatorists and complexity theoretists (and ourselves) that proof complexity is a little bit more than just the Hilbert-style game with abstract symbols on a sheet of paper.

7. Acknowledgment

We are grateful to Paul Beame, Stasys Jukna and the anonymous referee for a few corrections.

References

[BP96] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the* 37*th IEEE FOCS*, pages 274–282, 1996.

[Bla37] A. Blake. *Canonical expressions in Boolean algebra.* PhD thesis, University of Chicago, 1937.

[BC82] A. Borodin and S. Cook. A time–space trade-off for sorting on a general sequential model of computation. *SIAM J. on Computing*, 11:287–297, 1982.

[Bus87] S. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[BuP96] S. Buss and T. Pitassi. Resolution and the weak pigeonhole principle. Manuscript, 1996.

[BT88] S. Buss and G. Turán. Resolution proofs of generalized pigeonhole principle. *Theoretical Computer Science*, 62:311–317, 1988.

[Ch77] V. Chvátal. Determining the stable set number of a graph. *SIAM J. on Computing*, 6:1–14, 1977.

[CS88] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[DP60] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):210–215, 1960.

[Haj61] G. Hajós. Über eine Konstruktion nicht n-färbbarer Graphen. *Wiss. Zeitschr. Martin Luther Univ. Halle-Wittenberg*, 10:116–117, 1961.

[Hak85] A. Haken. The intractability or resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[Juk97] S. Jukna. Exponential lower bounds for semantic resolution. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetics: DIMACS workshop, April* 21-24, 1996*, DIMACS Series in Dicrete Mathematics and Theoretical Computer Science, vol.* 39, pages 163–172. American Math. Soc., 1997.

[Kra94] J. Krajíček. *Bounded arithmetic, propositional logic and complexity theory.* Cambridge University Press, 1994.

[LNNW95] L. Lovasz and I. Newman and M. Naor and A. Wigdreson. Search Problems in the Decision Tree Model *SIAM J. on Discrete Math.*, 8:119–132, 1995.

[Mc79] C. McDiarmid. Determining the chromatic number of a graph. *SIAM J. on Computing*, 8:643–662, 1979.

[Oko91] Е. А. Окольнишникова. Нижние оценки сложности реализации характеристических функций двоичных кодов бинарными программами. Методы дискретного анализа, 51:61–83, 1991. E. A. Okolnishnikova, Lower bounds for branching programs computing characteristic functions of binary codes, *Metody discretnogo analiza*, 51(1991), pages 61–83 (in Russian).

[PU92] T. Pitassi and A. Urquhart. The complexity of the Hajós calculus. In *Proceedings of the* 33*rd IEEE Symposium on Foundations of Computer Science*, pages 187–196, 1992.

[Raz91] A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *Proceedings of the 8th FCT, Lecture Notes in Computer Science,* 529, pages 47–60, New York/Berlin, 1991. Springer-Verlag.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[Tse68] Г. С. Цейтин. О сложности вывода в исчислении высказываний. In А. О. Слисенко, editor, *Исследования по конструктивной математике и математической логике,* II*; Записки научных семинаров ЛОМИ, т. 8*, pages 234–259. Наука, Ленинград, 1968. Engl. translation: G. C. Tseitin, On the complexity of derivations in propositional calculus, in: *Studies in mathematics and mathematical logic, Part II*, ed. A. O. Slissenko, pp. 115-125.

[Urq87] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

[Weg87] I. Wegener. *The complexity of Boolean functions.* Wiley-Teubner, 1987.