Proof Complexity of Pigeonhole Principles

Alexander A. Razborov [⋆]

Steklov Mathematical Institute, Moscow, Russia
Institute for Advanced Study, Princeton, USA

Abstract. The pigeonhole principle asserts that there is no injective mapping from m pigeons to n holes as long as $m > n$. It is amazingly simple, expresses one of the most basic primitives in mathematics and Theoretical Computer Science (counting) and, for these reasons, is probably the most extensively studied combinatorial principle. In this survey we try to summarize what is known about its proof complexity, and what we would still like to prove. We also mention some applications of the pigeonhole principle to the study of efficient provability of major open problems in computational complexity, as well as some of its generalizations in the form of general matching principles.

1 Introduction

Propositional proof complexity is an area of study that has seen a rapid development over a couple of last decades. It plays 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. Propositional proof complexity is in a sense complementary to the (non-uniform) computational complexity; moreover, there exist extremely rich and productive relations between the two areas. Besides the theory of (first-order) feasible proofs and computational complexity, propositional proof complexity is tightly connected in many ways with other areas like automated theorem proving and cryptography, and these connections are numerous both at the level of motivations and when it comes to proof techniques.

In my talk at the conference I gave a general overview of some important concepts, ideas and proof techniques behind this fascinating theory. A substantial portion of that material was already covered in various surveys and monographs (see e.g. [1–5]). For one particular subject from my lecture, however, the situation is very different. I am talking about the research on the proof complexity of specific tautologies that express various forms of the so-called *pigeonhole principle*. This principle (asserting that there is no injective mapping from m pigeons to n holes whenever $m > n$) is probably the most extensively studied combinatorial principle in proof complexity. It is amazingly simple and at the same time captures one of the most basic primitives in mathematics and Theoretical Computer Science (counting). It might be for these reasons that the pigeonhole principle somehow manages to find itself in the center of events, and many other

[⋆] Supported by The von Neumann Fund

important principles studied in the proof complexity are related to it in one or another way.

Surprisingly, the proof complexity of the pigeonhole principle essentially depends on the number of pigeons m (as the function of the number of holes n) and on subtle details of its representation as a propositional tautology. This leads to a rich structural picture, and results making the skeleton of this picture are amongst the most beautiful and important in the whole theory. Moreover, for a long time they have been determining its methods, machinery and ideology.

In the last couple of years several more important touches have been added to this picture, and these results do not seem to have been surveyed in the literature yet. For this reason, this written contribution is entirely devoted to the pigeonhole principle. It is organized as follows. In Section 2 we present the proof systems appearing in our survey. Our pace in this section will be slow, as it is primarily designed for the beginners; more experienced readers may take notice of the notation introduced there and skip all the rest. Section 3 contains formal definitions of the basic pigeonhole principle and its various modifications. The next section 4 is central: we give a survey of known lower and upper bounds on the proof complexity of the pigeonhole principle. With the help of some of these bounds, we show in Section 5 that such proof systems as Resolution and Polynomial Calculus do not possess efficient proofs of circuit lower bounds. In Section 6 we survey a few known results about the complexity of more general matching principles. Finally, we conclude with several open problems in Section 7.

2 Propositional Proof Systems

Let $x_1, \ldots, x_n, \ldots$ be propositional variables, and $\mathcal{C}$ be a certain class of propositional formulas in these variables. Denote by $\mathrm{TAUT}_{\mathcal{C}}$ the set of all tautologies in the class $\mathcal{C}$.

Informally speaking, a propositional proof system is any complete and sound calculus for generating members of $\mathrm{TAUT}_{\mathcal{C}}$. Given such a calculus, we can associate with it the "theorem-extracting" function P that for every binary string w encoding a legitimate proof in this calculus extracts the theorem $P(w)$ this proof proves.[1] Note that $\mathrm{im}(P) = \mathrm{TAUT}_{\mathcal{C}}$ (this is tantamount to saying that our calculus is complete and sound). Also, it is conceivable that for any reasonable calculus the function P will be polynomial time computable.

Cook and Reckhow [6] proposed to take these properties of the theorem-extracting function as a general *axiomatic definition* of a propositional proof system.

Definition 1 ([6]). *A* **propositional proof system** *(often abbreviated as p.p.s.) for a class $\mathcal{C}$ of propositional formulas is any polynomial time computable*

[1] If w is a string that does not make sense, we let $P(w)$ be any fixed tautology from $\mathrm{TAUT}_{\mathcal{C}}$.

function

$$P : \{0,1\}^* \overset{\text{onto}}{\longrightarrow} \mathrm{TAUT}_{\mathcal{C}}.$$

For a tautology $\phi \in \mathrm{TAUT}_{\mathcal{C}}$, any string w such that $P(w) = \phi$ is called a P- **proof of** *ϕ.*

Denote by $S_P(\phi)$ the minimal possible bit size $|w|$ of a P-proof w of ϕ.[2] Then the basic question of proof complexity can be formulated as simply as that: what can we say about $S_P(\phi)$ for "interesting" proof systems P and "interesting" tautologies ϕ?

Definition 1 paves way for relating propositional proof complexity to the rich structure developed in *computational* complexity. For example, let us call a p.p.s. P *p-bounded* if $S_P(\phi)$ is bounded from above by a polynomial in $|\phi|$. Then we have

Theorem 1 ([6]). $\mathbf{NP} = co-\mathbf{NP}$ *if and only if there exists a p-bounded propositional proof system.*

This easy result actually indicates that in its full generality Definition 1 is just a reformulation of the standard characterization of $co-\mathbf{NP}$ and has only little to do with the "real" proof theory. Propositional proof complexity really begins as a separate subject only when we are interested in the performance of concrete p.p.s. that are natural and that are of independent interest. Still, even in that case some standard structural concepts from computational complexity are extremely useful. As an example, let us introduce the following notion.

Definition 2. *A p.p.s. P is* p**-simulated** *by another p.p.s Q for the same class $\mathcal{C}$ if there exists a polynomial time algorithm A that transforms every P-proof into a Q-proof of the same tautology. That is, we demand that $\forall w \in \{0,1\}^*(Q(A(w)) = P(w))$. This relation is reflexive and transitive. Hence we say that two p.p.s are* p**-equivalent** *if they p-simulate each other.*

The notion of p-simulation is used for comparing strength of different proof systems (clearly, if P is p-simulated by Q then $S_Q(\phi) \leq S_P(\phi)^{O(1)}$) and arranging them into a hierarchy.

After this necessary digression into general structural issues, let us take a closer look at the specific proof systems used in this survey. The most basic example of a p.p.s. is the *Frege proof system* which is essentially the ordinary Hilbert-style propositional calculus from your favourite textbook in mathematical logic. Fix any complete language of propositional connectives (say, $L_0 \overset{\text{def}}{=} \{\neg, \wedge, \vee\}$), and let $\mathcal{C}$ consist of all propositional formulas in this language. A Frege proof system is specified by finitely many *Frege rules* of the form

$$\frac{A_1, \ldots, A_k}{B}\ (A_1, \ldots, A_k, B \in \mathcal{C}),$$

[2] It should be noted that although this complexity measure is by far more important, sometimes researchers are interested in more sophisticated ones. We will see below one example, degree of algebraic proofs.

called *axiom schemes* if $k = 0$. All these rules are required to be (implicationally) sound. A *Frege proof* consists of a sequence of applications of *substitutional instances* of Frege rules, defined as the result of substituting arbitrary formulas for variables. Finally, we demand that there are sufficiently many Frege rules meaning that the resulting Frege system must be (implicationally) complete.

At the first glance, this definition is rather arbitrary since we have a considerable freedom in choosing the set of Frege rules. It turns out, however, that the resulting Frege systems are all p-equivalent. More generally, Reckhow [7] proved the following (highly non-trivial) result.

Theorem 2 ([7]). *Let P, Q be two Frege systems considered as proof systems for tautologies in the language that consists of their common connectives. Then P and Q are p-equivalent.*

Thus, the Frege proof system is uniquely defined up to p-equivalence, and we jubilantly denote it by F. In fact, its definition is even more robust. Instead of a Hilbert-style calculus we may consider a Gentzen-style sequent calculus, and we will still get a proof system p-equivalent to F.

One more important remark to be made along these lines (robustness) is this. It is of absolutely no importance in the classical proof theory whether we represent proofs in a *tree-like* form or in the *sequential* (sometimes also called *dag-like*) form, when a proof is a sequence of formulas (sometimes called *lines*) in which every line is obtained from preceding lines via an inference rule. The situation is potentially very different in proof complexity: when we expand a sequential proof into a tree, the bit size may in general grow exponentially as every formula in the proof will repeat itself many times. It turns out, however, that this distinction does not matter for the Frege proof system, and its tree-like and sequential versions are p-equivalent.

All other propositional proof systems considered in this survey will be p-simulated by F (in fact, they will be even defined as Frege proofs of a special form). They will be represented in the form of a Gentzen-style sequent calculus, and they will be given in the sequential (as opposed to tree-like) form. Also, the propositional language will be from now on fixed to $L_0 \stackrel{\text{def}}{=} \{\neg, \wedge, \vee\}$. The connectives $\wedge$ and $\vee$ will be allowed to have an arbitrary number of arguments.[3] The (logical) *depth* of a propositional formula ϕ and the hierarchy Σ_d, Π_d of propositional formulas are defined in the standard way.

Fix any constant $d \geq 1$, and let $\mathcal{C} \stackrel{\text{def}}{=} \Sigma_d$. Every formula in $\mathcal{C}$ can be re-written as a sequent $A_1, \ldots, A_k \longrightarrow B_1, \ldots, B_\ell$ in which all formulas $A_1, \ldots, A_k, B_1, \ldots, B_\ell$ are of depth at most $d - 1$. The system F_d, by definition, is the fragment of F that operates with sequents of this form (that is, in which every formula is in $\Sigma_{d-1} \cup \Pi_{d-1}$!) and has all ordinary rules of a Gentzen-style calculus (structural rules, logical rules and the cut rule). With a slight abuse of

[3] The attentive reader might have observed at this point that we did not specify this parameter while discussing general Frege systems above. The reason should be already clear by now: the two versions are p-equivalent.

notation, when one is not interested much in the exact value of the depth d, the term *bounded-depth Frege proofs* is used.

Remark 1. One of the main ingredients in the proof of Theorem 2 is to show that every Frege proof of size S can be transformed into another Frege proof of size $S^{O(1)}$ and logical depth $O(\log S)$. In particular (with the same abuse of notation), the systems F and $F_{O(\log n)}$ have the same polynomial size proofs.

Remark 2. The p-simulation of sequential Frege proofs by tree-like Frege to a certain extent can be carried over to the bounded-depth case. Namely, F_d is p-simulated by tree-like F_{d+2} and, in fact, even by the tree-like version of some "symmetrized" variant of F_{d+1} in which we also allow Π_{d+1}-formulas (we did not make a special provision for Π_d-formulas in our definition of the *sequential* version of F_d as in that case they can be always w.l.o.g. broken up into a sequence of Σ_{d-1}-formulas).

Remark 3. Given the convenience of this representation in the Gentzen form, some authors even define the depth of a proof directly in terms of the sequent calculus (see e.g. [8]). It is important to remember that our notation (which reflects the semantical meaning of lines in a proof rather than the way they are syntactically represented) is off by one, and our F_d corresponds to depth $(d-1)$ sequent calculus proofs in their notation.

The system F_1 is called *Resolution* and denoted by R; this is one of the most important and frequently studied propositional proof systems. At the first glance there is something wrong about F_1 since it operates with Σ_1-formulas, and this class does not contain any non-trivial tautologies at all. This is easily circumvented by the following trick. We in fact prove tautologies in Σ_2 (i.e., in the *DNF form*), but instead of directly *proving* a tautology ϕ, we are *refuting* its negation $\bar{\phi} \in \Pi_2$. $\bar{\phi}$ is of the form $C_1 \wedge C_2 \wedge \ldots \wedge C_s$, where C_i are Σ_1 formulas (often called *clauses*), and the resolution proof system is trying to infer a contradiction (= the empty clause) from the set of axioms $\{C_1, C_2, \ldots, C_s\}$ operating with clauses only. By inspecting the rules of the Gentzen calculus, we see that the only non-trivial rule left in this situation is the atomic cut rule

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D}$$

that receives the special name *resolution rule.*

Let Σ_d^t, Π_d^t consist of those formulas in Σ_{d+1}, Π_{d+1} respectively for which the number of inputs of any connective at the bottom (closest to the variables) level is at most t. Clearly, $\Sigma_d \subseteq \Sigma_d^t \subseteq \Sigma_{d+1}$, and this intermediate class turned out to be very convenient in circuit complexity e.g. in the proof of *Håstad Switching Lemma* [9]. No wonder that it very naturally appears in the proof complexity, too. With a slight abuse of notation, we define $F_{d+0.5}$ as the fragment of F_{d+1} in which all occurring formulas are in fact in $\Sigma_d^{polylog(n)}$ (that is, in the corresponding sequents all formulas must be in $\Sigma_{d-1}^{polylog(n)} \cup \Pi_{d-1}^{polylog(n)}$), n the number of variables.

$F_{1.5}$ is in particular very close to resolution: it operates with sequents in which every formula is either a clause of polylogarithmic *width* (defined as the number of literals in the clause) or its negation (*elementary conjunction*). More generally, let $R(t)$ be the similarly defined extension of Resolution in which all formulas are in Σ_1^t (thus, $F_{1.5} = R(polylog)$). In this survey we will be particularly interested in $R(2)$ and $R(O(1))$.

Our last system is of completely different nature (at least, at the first glance). Fix any field $\mathbb{F}$, and let us interpret the logical constants TRUE and FALSE as the elements 0 and 1 in this field, respectively. Then the clause[4] $x_{i_1}^{\epsilon_1} \vee \ldots \vee x_{i_w}^{\epsilon_w}$ is satisfied by a truth assignment if and only if the corresponding 0-1 vector satisfies the polynomial equation $(x_{i_1} - \epsilon_1) \cdot \ldots \cdot (x_{i_w} - \epsilon_w) = 0$. We already saw before that proving Σ_2-formulas (DNFs) is equivalent to showing that a set of clauses is unsatisfiable, and now we take one step further and replace this task by the task of proving that the system of polynomial equations constructed from this set of clauses as described above does not have 0-1 solutions.

Polynomial Calculus (introduced in [10] and sometimes abbreviated as PC) is specifically designed for this latter task; it operates with polynomial equations over the field $\mathbb{F}$ and has *default axioms* $x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0$ (ensuring that x_is take on 0-1 values) and two inference rules

$$\frac{f = 0 \qquad g = 0}{\alpha f + \beta g = 0}\ (\alpha, \beta \in \mathbb{F}) \qquad\qquad \frac{f = 0}{f \cdot g = 0}.$$

The purpose is to infer the contradiction $1 = 0$ from a given system of polynomial equations. The complexity of a polynomial calculus proof is traditionally measured by its *degree* (defined as the maximal degree of all polynomials occurring in it) rather than by the bit size. Note that the default axioms allow us to assume w.l.o.g. that all polynomials in the proof are multi-linear; in particular, there always exists a polynomial calculus proof of degree $\leq n$.

3 Pigeonhole Principle(s)

Denote $\{1, \ldots, t\}$ by $[t]$. Let $m > n$ and $\{x_{ij} \mid i \in [m],\ j \in [n]\}$ be propositional variables that will be called *pigeonhole variables*. The *basic pigeonhole principle* PHP_n^m is the following DNF:

$$PHP_n^m \equiv \bigvee_{i \in [m]} \bigwedge_{j \in [n]} \bar{x}_{ij} \vee \bigvee_{j \in [n]} \bigvee_{\substack{i_1, i_2 \in [m] \\ i_1 \neq i_2}} (x_{i_1 j} \wedge x_{i_2 j}).$$

It will be more convenient (and, as we remarked in the previous section, simply necessary when working with p.p.s. below F_2) to work with its negation that

[4] we use here the convenient notation $x^1 \overset{\text{def}}{=} x$ and $x^0 \overset{\text{def}}{=} \bar{x}$

consists of the following groups of clauses ("axioms"):

$$Q_i \overset{\text{def}}{=} \bigvee_{j=1}^{n} x_{ij} \ \ (i \in [m]); \tag{1}$$

$$Q_{i_1,i_2;j} \overset{\text{def}}{=} (\bar{x}_{i_1 j} \vee \bar{x}_{i_2 j}) \ (i_1 \neq i_2 \in [m],\ j \in [n]). \tag{2}$$

These clauses express that a multi-valued mapping from $[m]$ ("pigeons") to $[n]$ ("holes") is both total (everywhere defined) and injective. Since no such mapping exists whenever $m > n$, PHP_n^m is indeed a tautology.

For the purpose of orientation let us see what will happen when the number of pigeons m increases. In that case the task of the propositional proof system becomes easier (unnecessary pigeons i can be just ignored), and the complexity in general may decrease. In other words, the principle becomes *weaker* (= easier to prove). If we are proving upper bounds on its complexity (work along with the proof system), our task is also easier, and lower bounds, on the contrary, are harder since we are fighting on the other side.

Many (if not all) results surveyed in the next section can be presented, if desired, as a smooth function in the two parameters m and n. It is more instructive, however, to let n tend to infinity and view m as a certain function in n. Then it turns out that there are several "critical points" at which most of the *qualitative* changes in the behaviour of the pigeonhole principle occur, and these are

$$m = n + 1, 2n, n^2, \infty.$$

For this reason, our tour in the next section will make a stop only at these four distinguished points, and we will see how drastically will the landscape be changing while we are driving from one to another.

The term "weak pigeonhole principle" traditionally refers to the case $m \geq 2n$. In order to distinguish between the three degrees of weakness, we use terms *moderately weak* ($m = 2n$), *weak* ($m = n^2$) and *very weak* ($m = \infty$).

As noted above, the "basic" pigeonhole principle expresses the fact that there is no multi-valued total injective mapping from $[m]$ to $[n]$. Besides increasing the number of pigeons, another way of weakening this principle consists in adding optional axioms dual to either (1) or to (2) or both. Namely, let

$$Q_{i;j_1,j_2} \overset{\text{def}}{=} (\bar{x}_{ij_1} \vee \bar{x}_{ij_2}) \ (i \in [m],\ j_1 \neq j_2 \in [n]) \tag{3}$$

(this group of axioms additionally requires that the assumed mapping is actually an ordinary single-valued function), and

$$Q_j \overset{\text{def}}{=} \bigvee_{i=1}^{m} x_{ij} \ \ (j \in [n]) \tag{4}$$

(which requires that the mapping is onto). The pigeonhole principle with the axioms (3) present is called *functional*, and the axioms (4) supply the prefix "*onto*" to the name. Altogether, we have four possible versions: the "basic" version PHP_n^m, the *functional version* $FPHP_n^m$, the *onto version* $onto - PHP_n^m$ and the *functional onto version* $onto - FPHP_n^m$. All these versions have been frequently studied in the literature, quite often interchangeably and sometimes confusingly, so the word "basic" should not be understood as an attempt to distinguish this particular version out in its family as the most important. Anyway, the thumb rule is the same as with the dependence of m on n: the longer the name, the weaker is the principle, the easier are upper bounds, and the harder are lower bounds.

In many cases the fact that the axioms (1) have large width is rather annoying (see e.g. Footnote 5 below). One more version of the pigeonhole principle called *extended pigeonhole principle* $EPHP_n^m$ was introduced in [11] as a convenient way of circumventing this. In order to avoid ambiguity in the original definition of $EPHP_n^m$, we present it here in more invariant framework of [12].

Namely, for every Boolean function $f(x_1, \ldots, x_n)$ in n variables and every pigeon $i \in [m]$ introduce a new *extension variable* y_{if}, and identify the pigeonhole variable x_{ij} with y_{i,x_j}. Then $EPHP_n^m$ is obtained by replacing the axioms (1) with new *local extension axioms*

$$y_{if_1}^{\epsilon_1} \vee y_{if_2}^{\epsilon_2} \vee y_{if_3}^{\epsilon_3} \quad \left(f_1^{\epsilon_1} \vee f_2^{\epsilon_2} \vee f_3^{\epsilon_3} \geq \bigvee_{j=1}^{n} x_j \right). \qquad (5)$$

It is easy to see that (1) can be easily inferred from (5) (more generally, local extension axioms of any width can be easily inferred from the local extension axioms (5) of width 3). Thus, $EPHP_n^m$ is weaker than PHP_n^m. [13, Section 5] observed that $FPHP_n^m$ is *weaker* than $EPHP_n^m$: namely, the substitution

$$y_{if}^{\epsilon} \mapsto \bigvee \{x_{ij} \mid f(\chi_j) = \epsilon\}$$

(where χ_j is the n-bit Boolean input with the only one in position j) transforms proofs of $EPHP_n^m$ into proofs of $FPHP_n^m$. The same argument shows that the naturally defined version $EFPHP_n^m$ (obtained by relaxing the condition in (5) to $f_1^{\epsilon_1} \vee f_2^{\epsilon_2} \vee f_3^{\epsilon_3} \geq \bigvee_{j=1}^{n} x_j \wedge \bigwedge_{j_1 \neq j_2} (\bar{x}_{j_1} \vee \bar{x}_{j_2})$) is in fact equivalent to $FPHP_n^m$.

The last subtle point is that our previous observation "m is larger $\Longrightarrow$ principle is weaker" is *no longer a priori valid* in the presence of the onto axioms (4). The reason is very simple: when we restrict our mapping to fewer pigeons, as one of the by-side results of this restriction, the axioms (4) can get falsified.

4 Survey of results

In this section we give a survey of at least the most important results about the complexity of the pigeonhole principle. As we promised in the previous section, our tour will make four stops, $m = n + 1, 2n, n^2, \infty$. When presenting lower bounds, we always try our best to identify the weakest version (the strongest for upper bounds) to which the result is applicable, even if the original paper did not elaborate on the issue.

4.1 Classical case: $m = n + 1$ (hard but expected)

It was the pigeonhole principle for which the first resolution lower bounds were proven in [14]. Haken's result is viewed by many as *the* result that really started the area of propositional proof complexity off.

Theorem 3 **([14]).** $S_R(onto - FPHP_n^{n+1}) \geq \exp(\Omega(n))$.

Buss [15] further confirmed that Frege is a very powerful proof system by showing that the pigeonhole principle (any version) is easy for it.

Theorem 4 **([15]).** $S_F(PHP_n^m) \leq n^{O(1)}$.

One natural proper subsystem of Frege in which PHP_n^m still has polynomial size proofs was identified in [16]; it is called the *cutting planes* proof system.

The next major step was to analyze the complexity of the pigeonhole principle with respect to bounded-depth Frege proof systems. Ajtai [17] proved a superpolynomial lower bound using non-standard models of arithmetic, and Bellantoni, Pitassi and Urquhart [18] presented a combinatorial version of his argument, at the same time extracting from it the following explicit bound:

Theorem 5 **([17, 18]).** *For every fixed $d > 0$, $S_{F_d}(onto - FPHP_n^{n+1}) \geq n^{\epsilon_d \log n}$, where ϵ_d is a constant depending only on d.*

Finally, Pitassi, Beame, Impagliazzio [19] and, independently, Krajíček, Pudlak and Woods [20] improved this lower bound to truly exponential.

Theorem 6 **([19, 20]).** *For every fixed $d > 0$, $S_{F_d}(onto - FPHP_n^{n+1}) \geq \exp(n^{\epsilon_d})$, where ϵ_d is a constant depending only on d.*

Another proof system for which PHP was the first tautology to be shown to be hard is Polynomial Calculus.[5] The following lower bound proved by Razborov [21] is applicable to an arbitrarily weak pigeonhole principle.

Theorem 7 **([21]).** *Every polynomial calculus proof of $FPHP_n^\infty$ (over an arbitrary field $\mathbb{F}$) must have degree $\Omega(n)$.*

The proof of Theorem 7 uses a rather specific combinatorial argument called *pigeon dance*. The original proof was somewhat simplified in [22], although even that simpler form essentially depended on the pigeon dance. [23] gave another proof of the same $\Omega(n)$ lower bound which almost immediately follows from some general theory, but it can be applied only to the stronger principle $EPHP_n^m$ and only when $m = O(n)$ (cf. Theorem 29 below).

[5] One has to be a little bit creative when translating the axioms (1) to the algebraic language since the straightforward translation from Section 2 produces polynomials of intolerably high degree n. Instead, we transform them into degree 1 equations $\sum_{j\in[n]} x_{ij} - 1 = 0$. An alternative (and equivalent) way is to consider at once the extended version $EFPHP_n^m$ mentioned in Section 3.

Looking at the statement of Theorem 7 more closely, we see that unlike all other negative results in this section, the prefix "onto" is missing there. The following observation made by Riis [24] shows that there is a very good reason for this omission.

Theorem 8 ([24]). *Assume that the ground field $\mathbb{F}$ has characteristic p and that $\binom{m}{d} \not\equiv \binom{n}{d} \pmod{p}$ for some $d \geq 1$. Then $onto - PHP_n^m$ has degree d polynomial calculus proof over $\mathbb{F}$.*

If $m = n{+}p^\ell$ then $\binom{m}{d} \equiv \binom{n}{d} \pmod{p}$ for all $d < \min\{n{+}1, p^\ell\}$ and Theorem 8 is no longer applicable, say, when n and p^ℓ are of the same order. Moreover, there are good reasons to believe (see Question 7 in Section 7) that $onto - PHP_n^{n+p^\ell}$ is hard for the polynomial calculus in characteristic p. Therefore, Theorem 8 perfectly illustrates the point made in Section 3: for the onto version, we no longer can assume that the complexity will be anti-monotone in m; instead, it may oscillate, at least for algebraic proof systems.

4.2 Moderately Weak PHP_n^m: $m = 2n$ (mystery begins)

Many results in Section 4.1 are hard and deep but all of them were a sort of expected. It will be just the opposite in this section (I mean of course only the last part, their hardness and depth are quite competitive).

There is no a priori reason to believe that Theorem 6 can not be generalized to larger values of m. Nonetheless, it is indeed the case, and in fact this had been first shown even *prior* to that theorem. Namely, Paris, Wilkie and Woods proved in [25] that $FPHP_n^{2n}$ *does* possess short bounded-depth proofs, and Krajíček [26] calculated the exact value of depth resulting from their proof.

Theorem 9 ([25, 26]).

a) $S_{F_{2.5}}(FPHP_n^{2n}) \leq n^{O(\log n)}$;
b) $S_{F_{1.5}}(onto - FPHP_n^{2n}) \leq n^{O(\log n)}$.

Buss and Turán [27] showed that the situation with Theorem 3 is exactly the opposite (at this stop!), and it readily generalizes to the moderately weak principle.

Theorem 10 ([27]). $S_R(onto - FPHP_n^{2n}) \geq \exp(\Omega(n))$.

In another important development, Maciel, Pitassi and Woods [8] were able to generalize Theorem 9 b) (and improve Theorem 9 a)) to the case of the basic principle:

Theorem 11 ([8]). $S_{F_{1.5}}(PHP_n^{2n}) \leq n^{O(\log n)}$.

Finally, the following recent result by Atserias [28] shows that in this situation depth can be traded for size.

Theorem 12 ([28]). $S_{F_d}(PHP_n^{2n}) \leq n^{((\log n)^{O(1/d)})}$.

Theorems 10 and 11 still leave open the gap between $R = R(1)$ and $F_{1.5} = R(polylog)$. All our intuition from complexity theory and mathematical logic strongly suggests that the distance between 2 and polylog should be much shorter than between 2 and 1. The last mysterious result (in Section 4.2!) due to Atserias, Bonet and Esteban [29] indicates that for the pigeonhole principle the situation is exactly the opposite.

Theorem 13 ([29]). $S_{R(2)}(onto - FPHP_n^{2n}) \geq \exp(n/(\log n)^{O(1)})$.

The case of $R(3)$ is still open.

4.3 Weak PHP_n^m: $m = n^2$ (mystery becomes hard labour)

Theorem 12 gets significantly improved in this case. Like Theorem 9, this was explicitly extracted from the paper [25] by Krajíček [2].

Theorem 14 ([25, 2]). $S_{F_d}(PHP_n^{n^2}) \leq n^{(\log^{(\Omega(d))} n)}$*, where* $\log^{(t)}$ *is the* t*-wise composition of* $\log$ *with itself.*

The most remarkable thing that happens at this stop, however, is that the proof method of Theorems 3, 10 also completely breaks down. The question on determining the resolution proof complexity for the weak PHP has been very intriguing since it became clear, and it has been solved only very recently. Section 4.3 is entirely devoted to the history of this new result, which can be essentially viewed as the history of accumulating the necessary techniques.

The first major contribution was made by Buss and Pitassi [30]. Firstly, they proposed an extremely convenient "normal form" for resolution proofs of the pigeonhole principle that was used in many subsequent papers on the subject (proofs in this normal form operate exclusively with *monotone*, i.e., negation-free clauses). As a rather surprising corollary of this normal form, they showed that PHP_n^m and $onto - PHP_n^m$ behave in the same way with respect to Resolution.

Theorem 15 ([30]).
$S_R(onto - PHP_n^m) \leq S_R(PHP_n^m) \leq S_R(onto - PHP_n^m)^{O(1)}$.

This phenomenon seems to be very unique: Theorem 7 and Theorem 8 in particular imply that this is certainly not the case for the polynomial calculus.

Secondly, [30] solved the (relatively easy) case of tree-like Resolution.

Theorem 16 ([30]). *Every tree-like resolution proof of* $onto - FPHP_n^m$ *must have size at least* 2^n*, for any* $m > n$.

The next contribution was made in the paper by Razborov, Wigderson and Yao [31]. They identified two subsystems of Resolution and proved the desired lower bound for each of them (using different methods). We mention here only one of these systems, *rectangular calculus*, although we skip its formal definition. Intuitively, the idea is to concentrate only on those monotone clauses that are of "rectangular shape" $\bigvee_{i \in I} \bigvee_{j \in J} x_{ij}$ with $I \subseteq [m]$, $J \subseteq [n]$, and write down appropriate sound rules for operating with such clauses.

Theorem 17 ([31]). *Every rectangular calculus proof of $FPHP_n^{n^2}$ must have size* $\exp(\Omega(n/(\log n)))$.

A resolution proof is called *regular* if along every path in this proof every literal is resolved at most once. Proofs in every one of the two subsystems of Resolution considered in [31] are in fact regular. The following result by Pitassi and Raz made a major improvement on [31]:

Theorem 18 ([32]). *Every regular resolution proof of $FPHP_n^{n^2}$ must have size* $\exp(n/(\log n)^{O(1)})$.

Shortly after Raz [33] came up with a complete solution for the basic version PHP_n^m. By Theorem 15, this immediately extends to the onto version.

Theorem 19 ([33]). $S_R(onto - PHP_n^{n^2}) \geq \exp(n/(\log n)^{O(1)})$.

Razborov [13] gave a simpler proof of the same result. In the next paper [34] the lower bound was extended to the functional case, and, finally, in [35] the weakest functional onto version was also analyzed.

Theorem 20 ([13, 34, 35]). $S_R(onto - FPHP_n^{n^2}) \geq \exp(\Omega(n/(\log n)^2))$.

4.4 Very Weak PHP: $m = \infty$ (last twinkle of mystery)

The reader who feels uncomfortable with infinitely many pigeons, may think of m in this section as of a sufficiently large number.

One more surprise still awaits us at this last stop. Namely, it had been conjectured for a while that $S_R(PHP_n^m) = \exp(\Omega(n))$ for every m whatsoever, even if we are not smart enough to prove this. The paper [30] already mentioned above disproved the conjecture, and [31] analyzed their proof to show that it is in fact carried over in the rectangular calculus.

Theorem 21 ([30]). *There exists a rectangular calculus proof of PHP_n^∞ that has size* $\exp(O((n \log n)^{1/2}))$.

All lower bounds from Section 4.3 readily extend to the very weak case (in fact, all of them were originally stated in this form).

Theorem 22 ([31]). *Every rectangular calculus proof of $FPHP_n^\infty$ must have size* $\exp(\Omega(n^{1/2}))$.

Theorems 21 and 22 determine the rectangular calculus complexity of the very weak pigeonhole principle up to a logarithmic factor in the exponent.

Theorem 23 ([32]). *Every regular resolution proof of $FPHP_n^\infty$ must have size* $\exp(n^{\Omega(1)})$.

Theorem 24 ([33]). $S_R(onto - PHP_n^\infty) \geq \exp(n^{\Omega(1)})$.

[33] also estimates the constant assumed in the expression $n^{\Omega(1)}$ above as between 1/10 and 1/8.

Theorem 25 ([13, 34, 35]). $S_R(onto - FPHP_n^\infty) \geq \exp(\Omega(n^{1/3}))$.

5 Application: circuit lower bounds are hard for weak proof systems

As we mentioned in Introduction, one of the reasons for the popularity of the pigeonhole principle consists in its tight connections with many other things. This and the next sections illustrate the point.

In [36, Appendix] Razborov proposed to study the provability of (first-order or second-order) principles expressing in a particular way that a given Boolean function can not be computed by short Boolean circuits. Shortly after, J. Krajíček observed that this question possesses an adequate re-formulation in terms of propositional proofs (which is by far more convenient), and it is this framework that is followed here.

More specifically, let f_n be a Boolean function in n variables, and let $t \leq 2^n$. Denote by $Circuit_t(f_n)$ any natural CNF *of size* $2^{O(n)}$ encoding the description of a size-t fan-in 2 Boolean circuit presumably computing f_n. Then its negation is a tautology if and only if the circuit size of f_n is greater than t. We demand that every clause in $Circuit_t(f_n)$ is of constant width (5 is enough). Given our liberal $2^{O(n)}$ bound on the size of $Circuit_t(f_n)$, it can be easily constructed simply by introducing a separate propositional variable x_{av} for the Boolean value computed at the node v on the input string $a \in \{0,1\}^n$; for a precise definition see [21, 35]. Raz [33] also proposed to consider the variant of this principle in which *unbounded fan-in* circuits are allowed; we will denote this version by $Circuit_t^+(f_n)$. It is stronger than $Circuit_t(f_n)$, and we can no longer demand that every axiom is of constant width.

One of the main motivations for proposing this framework was that *all known lower bound proofs in circuit complexity can be carried over in it.* That is, if we restrict the class of circuits used in the definition of $Circuit_t(f_n)$ to monotone circuits, bounded-depth circuits, depth-2 threshold circuits etc. then this principle *will* become provable within polynomial (that is, $2^{O(n)}$) size in the Frege system or, in the worst case, in its natural extension known as *Extended Frege.* On the other hand, it is known that $Circuit_t(f_n)$ is hard for every proof system possessing *Efficient Interpolation Theorem* (see e.g. [4] for definitions and discussion of this theorem), but only *provided strong one-way functions exist.* In particular, Resolution and Polynomial Calculus do have Efficient Interpolation, hence the latter conclusion applies to them.

The only known *unconditional* (i.e., without any unproven assumptions) lower bounds on the complexity of $Circuit_t(f_n)$ are based on some of the lower bounds for the pigeonhole principle surveyed in the previous section, and on a reduction from PHP to $Circuit_t(f_n)$ discovered in [21]. The underlying idea of this reduction is simple. Let $A \overset{\text{def}}{=} f_n^{-1}(1)$. Then every counterexample to $onto-FPHP_t^{|A|}$ (encoded by the pigeonhole variables $\{x_{aj} \mid a \in A, j \in [t]\}$) can be used to construct a short circuits for f_n; namely, $f_n \equiv \bigvee_{j\in[t]} K_j$, where K_j is the characteristic function of that input $a \in A$ for which $x_{aj} = 1$. Elaborating a little bit on this simple idea, we can get rid of the onto axioms (in the case of PC!) and show

Lemma 1 ([21]). *If $Circuit_t(f_n)$ has polynomial calculus proof of degree d for some function f_n, then $FPHP^{2^n}_{t/2n}$ also has a PC proof of the same degree d.*

Lemma 1 and Theorem 7 immediately imply

Corollary 1 ([21]). *Every polynomial calculus proof of $Circuit_t(f_n)$ (for any function f_n) must have degree $\Omega(t/n)$.*

To extend the reduction from Lemma 1 to the case of Resolution, we apparently must employ the "onto" axioms (4).[6] Denote $|A|$ by m.

Lemma 2. **a) [33]** $S_R(Circuit^+_t(f_n)) \geq S_R(onto - PHP^m_{t-1})$;
b) [35] $S_R(Circuit_t(f_n)) \geq S_R(onto - FPHP^m_{t/2n})$.

Combining Lemma 2 a) with Theorem 24, we immediately get

Theorem 26 ([33]). $S_R(Circuit^+_t(f_n)) \geq \exp(t^{\Omega(1)})$.

Combining Lemma 2 b) with Theorem 25, we get a similar bound for $Circuit_t(f_n)$. If we, however, take into account that in Lemma 2 we always have $m \leq 2^n$, we can do slightly better (cf. Theorem 31 in the next section) and actually prove

Theorem 27 ([35]). $S_R(Circuit_t(f_n)) \geq \exp(\Omega(t/n^3))$.

6 Generalization: matching principles

One natural way to interpret $FPHP^m_n$ $[onto - FPHP^m_n]$ is by saying that the complete bipartite graph on two sets of vertices U and V with $|U| = m,\ |V| = n$ does not contain a matching from U to V [a perfect matching, respectively]. One very natural question is what can be said about the complexity of this (perfect) *matching principle* for other graphs, not necessarily bipartite, or, perhaps, even hypergraphs. In this section we will survey what is known along these lines.

Ben-Sasson and Wigderson [11] introduced the principle $G - PHP$ which says that a given bipartite graph G from U to V does not contain a multi-valued matching (thus, $PHP^m_n \equiv K_{m,n} - PHP$), and this definition readily generalizes to all other versions of the pigeonhole principle. They were able to relate the complexity of $G - FPHP$ to expansion properties of the graph G.

Theorem 28 ([11]). *For every bipartite graph G on (U, V) which is a constant-rate expander of bounded minimal degree*[7]*, $S_R(G - FPHP) \geq \exp(\Omega(|U|))$.*

Alekhnovich and Razborov [21] used a general theory developed in that paper to show that the expansion properties of G also imply hardness with respect to the polynomial calculus.

[6] The remark in [13, Section 5] that $FPHP^m_n$ would suffice for the purpose seems to be erroneous.

[7] the minimum is taken over the nodes in U

Theorem 29 **([21]).** *For every bipartite graph G on (U, V) which is a constant-rate expander of bounded minimal degree, every polynomial calculus proof of $G - EPHP$ must have degree $\Omega(|U|)$.*

For an arbitrary (not necessarily bipartite) graph G, let $PM(G)$ be the principle asserting that G does not contain a *perfect* matching (thus, $onto{-}FPHP_n^m \equiv PM(K_{m,n})$). As an example, let G be a $(2n \times 2n)$ grid with two opposite corners removed, then $PM(G)$ is called the *mutilated chessboard problem.* Dantchev and Riis [37] proved the following tight lower bound for it (independently, Alekhnovich [38] showed a somewhat weaker bound $\exp(\Omega(n^{1/2}))$):

Theorem 30 **([37]).** *Every resolution proof of the mutilated chessboard problem must have size* $\exp(\Omega(n))$.

Razborov [35] considered the principle $PM(G)$ in full generality. Let $\delta(G)$ be the minimal degree of a vertex $v \in V(G)$.

Theorem 31 **([35]).** $S_R(PM(G)) \geq \exp\left(\Omega\left(\frac{\delta(G)}{(\log |V(G)|)^2}\right)\right)$.

This is a far-going generalization of Theorems 20 and 25. Vice versa, it is worth noting that Theorem 31 is in fact proved by a sort of indirect reduction from $FPHP_n^m$, followed by applying methods from [34].

Finally, [35] also contains a further generalization of Theorem 31 to hypergraphs. We formulate it here only for the special case of *complete* r*-hypergraph* intensively studied in the literature. Namely, let $r \nmid n$ and the principle $Count_r^n$ assert that an n-element set can not be partitioned into r-sets.

Theorem 32 **([35]).** $S_R(Count_r^n) \geq \exp(\Omega(n/(r^2(\log n)(r + \log n))))$.

7 Open problems

1. Theorem 4 and Remark 1 imply that PHP_n^{n+1} has polynomial size proofs of logical depth $O(\log n)$, whereas Theorem 6 implies that no such proof exists in bounded depth. That would be nice to further narrow this gap. In particular, is it true that $S_{F_{O(\log\log n)}}(PHP_n^{n+1}) \leq n^{O(1)}$?
2. Is it true that for some absolute constant $d \geq 1$ and for some $m = m(n)$, $S_{F_d}(onto{-}FPHP_n^m) \leq n^{O(1)}$? Quasi-polynomial upper bounds are provided by Theorems 9, 11, 12, 14, with the latter result particularly close to a polynomial.
3. Is it true that for some absolute constant $t \geq 1$ and for some $m = m(n)$, $S_{R(t)}(onto - FPHP_n^m) \leq \exp((\log n)^{O(1)})$? This is not true if $t \leq 2$ (Theorem 13), but becomes true if t is allowed to grow polylogarithmically in n (Theorems 9 b), 11).
4. When $m \geq n^2$, we do not know how to answer the previous question even for $t = 2$. Is it true that $S_{R(2)}(onto - FPHP_n^{n^2}) \leq \exp((\log n)^{O(1)})$?
5. What is the value of $\limsup_{n\to\infty} \frac{\log_2 \log_2 S_R(PHP_n^\infty)}{\log_2 n}$? From Theorems 21 and 25 we know that it lies in the interval $[1/3, 1/2]$.

6. In order to make this survey more structured, we stopped only at $m = n+1, 2n, n^2, \infty$ to record the changes that had had happened along the road. The question of identifying the "turning points" more exactly also seems to be of considerable interest. As an example, consider the case $m = n+n^{1/2}$. It is consistent with our current knowledge that $S_{F_d}(PHP_n^{n+n^{1/2}}) \geq \exp(n^{\epsilon_d})$ for any fixed d, and it is also consistent that $S_{F_{1.5}}(PHP_n^{n+n^{1/2}}) \leq n^{O(\log n)}$. Rule out one of these two possibilities.
7. Is it true that for a fixed prime p and $\ell, n \to \infty$, $\mathit{onto}-FPHP_n^{n+p^\ell}$ does not possess bounded-degree PC proofs over any field of characteristic p? This is known for the weaker *Nullstellensatz proof system* [39].
8. Let G be a constant-rate bounded-degree expander on the sets of vertices U, V. Is it true that every polynomial calculus proof of $G-FPHP$ must have degree $\Omega(|U|)$? Theorem 29 answers this question for the stronger version $G - EPHP$.

8 Acknowledgement

I am grateful to Paul Beame and Sam Buss for useful suggestions promptly made on the original draft of this survey. Paul Beame has also contributed several open problems.

References

1. Urquhart, A.: The complexity of propositional proofs. Bulletin of Symbolic Logic **1** (1995) 425–467
2. Krajíček, J.: Bounded arithmetic, propositional logic and complexity theory. Cambridge University Press (1995)
3. Razborov, A.: Lower bounds for propositional proofs and independence results in Bounded Arithmetic. In auf der Heide, F.M., Monien, B., eds.: Proceedings of the 23rd ICALP, Lecture Notes in Computer Science, 1099, New York/Berlin, Springer-Verlag (1996) 48–62
4. Beame, P., Pitassi, T.: Propositional proof complexity: Past, present and future. Technical Report TR98-067, Electronic Colloquium on Computational Complexity (1998)
5. Pudlák, P.: The lengths of proofs. In Buss, S., ed.: Handbook of Proof Theory. Elsevier (1998) 547–637
6. Cook, S.A., Reckhow, A.R.: The relative efficiency of propositional proof systems. Journal of Symbolic Logic **44** (1979) 36–50
7. Reckhow, R.A.: On the lengths of proofs in the propositional calculus. Technical Report 87, University of Toronto (1976)
8. Maciel, A., Pitassi, T., Woods, A.: A new proof of the weak pigeonhole principle. Manuscript (1999)
9. Håstad, J.: Computational limitations on Small Depth Circuits. PhD thesis, Massachusetts Institute of Technology (1986)
10. Clegg, M., Edmonds, J., Impagliazzo, R.: Using the Groebner basis algorithm to find proofs of unsatisfiability. In: Proceedings of the 28th ACM STOC. (1996) 174–183

11. Ben-Sasson, E., Wigderson, A.: Short proofs are narrow - resolution made simple. In: Proceedings of the 31st ACM STOC. (1999) 517–526
12. Alekhnovich, M., Ben-Sasson, E., Razborov, A., Wigderson, A.: Pseudorandom generators in propositional complexity. In: Proceedings of the 41st IEEE FOCS. (2000) 43–53
13. Razborov, A.: Improved resolution lower bounds for the weak pigeonhole principle. Technical Report TR01-055, Electronic Colloquium on Computational Complexity (2001) Available at ftp://ftp.eccc.uni-trier.de/pub/eccc/reports/2001/TR01-055/index.html.
14. Haken, A.: The intractability or resolution. Theoretical Computer Science **39** (1985) 297–308
15. Buss, S.R.: Polynomial size proofs of the propositional pigeonhole principle. Journal of Symbolic Logic **52** (1987) 916–927
16. Cook, W., Coullard, C.R., Turán, G.: On the complexity of cutting plane proofs. Discrete Applied Mathematics **18** (1987) 25–38
17. Ajtai, M.: The complexity of the pigeonhole principle. In: Proceedings of the 29th IEEE Symposium on Foundations of Computer Science. (1988) 346–355
18. Bellantoni, S., Pitassi, T., Urquhart, A.: Approximation of small depth Frege proofs. SIAM Journal on Computing **21** (1992) 1161–1179
19. Pitassi, T., Beame, P., Impagliazzo, R.: Exponential lower bounds for the pigeonhole principle. Computational Complexity **3** (1993) 97–140
20. Krajíček, J., Pudlák, P., Woods, A.R.: Exponential lower bounds to the size of bounded depth Frege proofs of the pigeonhole principle. Random Structures and Algorithms **7** (1995) 15–39
21. Razborov, A.: Lower bounds for the polynomial calculus. Computational Complexity **7** (1998) 291–324
22. Impagliazzo, R., Pudlák, P., Sgall, J.: Lower bounds for the polynomial calculus and the Groebner basis algorithm. Computational Complexity **8** (1999) 127–144
23. Alekhnovich, M., Razborov, A.: Lower bounds for the polynomial calculus: non-binomial case. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. (2001) 190–199
24. Riis, S.: Independence in Bounded Arithmetic. PhD thesis, Oxford University (1993)
25. Paris, J.B., Wilkie, A.J., Woods, A.R.: Provability of the pigeonhole principle and the existence of infinitely many primes. Journal of Symbolic Logic **53** (1988) 1235–1244
26. Krajíček, J.: On the weak pigeonhole principle. Fundamenta Mathematicae **170** (2001) 123–140
27. Buss, S., Turán, G.: Resolution proofs of generalized pigeonhole principle. Theoretical Computer Science **62** (1988) 311–317
28. Atserias, A.: Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms. In J. Sgall, A. Pultr, P.K., ed.: Proceedings of the 26th International Symposium on the Mathematical Foundations of Computer Science (Marianske Lazne, August '01), Lecture Notes in Computer Science 2136, Springer-Verlag (2001) 148–158
29. Atserias, A., Bonet, M.L., Esteban, J.L.: Lower bounds for the weak pigeonhole principle beyond resolution. To appear in *Information and Computation* (2000)
30. Buss, S., Pitassi, T.: Resolution and the weak pigeonhole principle. In: Proceedings of the CSL97, Lecture Notes in Computer Science, 1414, New York/Berlin, Springer-Verlag (1997) 149–156

31. Razborov, A., Wigderson, A., Yao, A.: Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In: Proceedings of the 29th ACM Symposium on Theory of Computing. (1997) 739–748
32. Pitassi, T., Raz, R.: Regular resolution lower bounds for the weak pigeonhole principle. In: Proceedings of the 33rd ACM Symposium on the Theory of Computing. (2001) 347–355
33. Raz, R.: Resolution lower bounds for the weak pigeonhole principle. Technical Report TR01-021, Electronic Colloquium on Computational Complexity (2001)
34. Razborov, A.: Resolution lower bounds for the weak functional pigeonhole principle. Manuscript, available at http://www.mi.ras.ru/˜razborov/matching.ps (2001)
35. Razborov, A.: Resolution lower bounds for perfect matching principles. Manuscript, available at http://www.mi.ras.ru/˜razborov/matching.ps (2001)
36. Razborov, A.: Bounded Arithmetic and lower bounds in Boolean complexity. In Clote, P., Remmel, J., eds.: Feasible Mathematics II. Progress in Computer Science and Applied Logic, vol. *13*. Birkhaüser (1995) 344–386
37. Dantchev, S., Riis, S.: "Planar" tautologies hard for Resolution. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. (2001) 220–229
38. Alekhnovich, M.: Mutilated chessboard is exponentially hard for resolution. Manuscript (2000)
39. Beame, P., Riis, S.: More on the relative strength of counting principles. In Beame, P., Buss, S., eds.: Proof Complexity and Feasible Arithmetics: DIMACS workshop, April 21-24, 1996, DIMACS Series in Dicrete Mathematics and Theoretical Computer Science, vol. 39. American Math. Soc. (1997) 13–35