An Exponential Separation between the Matching Principle and the Pigeonhole Principle

Paul Beame *
Computer Science and Engineering
FR-35
University of Washington
Seattle, WA 98195
beame@cs.washington.edu

Toniann Pitassi †
Computer Science Department
UC San Diego
9500 Gilman Drive
La Jolla, CA 92093-0114
toni@cs.ucsd.edu

Abstract

The combinatorial matching principle states that there is no perfect matching on an odd number of vertices. This principle generalizes the pigeonhole principle, which states that for a fixed bipartition of the vertices, there is no perfect matching between them. Therefore, it follows from recent lower bounds for the pigeonhole principle that the matching principle requires exponential-size bounded-depth Frege proofs. Ajtai [Ajt90] previously showed that the matching principle does not have polynomial-size bounded-depth Frege proofs even with the pigeonhole principle as an axiom schema. His proof utilizes nonstandard model theory and is nonconstructive. We improve Atjai's lower bound from barely superpolynomial to exponential and eliminate the nonstandard model theory.

Our lower bound is also related to the inherent complexity of particular search classes (see [Pap91]). In particular, oracle separations between the complexity classes PPA and $PPAD$, and between PPA and PPP also follow from our techniques ([BP93a]).

1 Introduction

Recently, it has been shown that the propositional pigeonhole principle requires exponential-size, bounded-depth Frege proofs [BIK+92, KPW91, PBI]. It is natural to ask what new theorems can and cannot be proven in polynomial size, and bounded depth, if we allow the pigeonhole principle as an axiom schema.

The propositional pigeonhole principle can be expressed by a family of propositional formulas, $\{PHP_m : m \geq 0\}$, where PHP_m asserts that there is no 1-1 mapping from a set D_0 of size $m+1$ to a set D_1 of size m. A related, but more general principle is the *perfect matching principle*, NPM_n, which states that no graph on $2n+1$ nodes consists of a perfect matching. We encode NPM_n using $\binom{2n+1}{2}$ *matching* variables, $P_{\{i,j\}}$, $i,j \leq 2n+1$. Using these variables, NPM_n can be written as the disjunction of the following *matching clauses*:

$$\bigwedge\{\neg P_{\{i,j\}} : j \leq 2n+1,\ j \neq i\},\ i \leq 2n+1;$$
$$\bigwedge\{P_{\{i,k\}}, P_{\{j,k\}}\}, i \neq j \neq k,\ i,j,k \leq 2n+1.$$

It is not too hard to see that if there are short, bounded-depth Frege proofs of NPM_n, then there are also short, bounded-depth Frege proofs of the onto version of the pigeonhole principle. Expressed propositionally, the onto version of PHP has additional terms which imply that the function is also surjective. The recent exponential size lower bound for bounded-depth Frege proofs of the onto version of PHP [BIK+92] thus also establishes an exponential size lower bound for bounded-depth Frege proofs of NPM_n. (A proof of Urquhart [Urq87] shows that NPM_n, like PHP_n, does have a polynomial size Frege proof of logarithmic depth.)

This suggests the following question: is the matching principle strictly stronger than the pigeonhole principle? Ajtai [Ajt90] was the first to

*Research supported by NSF grants CCR-8858799 and CCR-8907960

†Research supported by an NSF postdoctoral fellowship

show that, in a precise sense, the matching principle is stronger than the pigeonhole principle. One can generalize the pigeonhole principle by allowing each variable in the PHP formula to represent an arbitrary *formula* over some underlying set of propositional variables. Now consider a bounded-depth Frege proof system, with underlying *matching* variables $P_{\{i,j\}}$, where the system is strengthened by allowing all bounded-depth instances of the PHP as axioms. In [Ajt90], Ajtai showed that NPM_n does not have polynomial-sized, bounded-depth Frege proofs, even in this stronger system.

The structure of Ajtai's argument extends the proof technique in his superpolynomial lower bound for the PHP [Ajt88]. He first sketches a restriction lemma giving small 'covering sets' for formulas over the matching variables. Then, in the novel part of the paper, he shows how the restrictions cannot have falsified the PHP axioms. This is done by first showing that all the information about a given pigeon or hole in a PHP axiom can be determined by the values of the matching variables touching a small covering set and then showing that it is not possible to have the information about pigeons or holes locally appear to describe a 1-1 function and yet be globally consistent. This last piece forms the bulk of the paper and uses a somewhat involved counting argument.

In this paper, we present a new proof and we improve the lower bound from superpolynomial to exponential. This result uses the proof-theoretic methods from [BIK+92, BPU91, KPW91] and a modification of the switching lemma from [BIK+92, PBI]. The most difficult new part of this proof is showing that each restricted PHP axiom is converted to an approximation of a true formula after the various conversions are made. The structure of this argument is similar to Ajtai's: As in [Ajt90] we use a bit-wise encoding of the PHP formulas to obtain small descriptions of what happens to each pigeon or hole. In our case, rather than small covering sets we use small height matching decision trees along the lines of [BIK+92, PBI]. This difference is fortuitous because it turns out that this permits a much simpler counting argument to show that it is impossible for the converted subformulas of the PHP axiom to locally describe a 1-1 function and be globally consistent.

The lemma which shows this latter result, is of independent interest. In particular, in [BP93a], we use it to demonstrate oracle separations between certain complexity classes of search problems: between classes PPA and $PPAD$ and between classes PPA and PPP. These complexity classes, which characterize the complexity of many interesting problems, were defined by Papadimitriou [Pap91] and lie between the function versions of P and NP.

In this extended abstract we emphasize the proof-theoretic aspects and mostly concentrate on places where there are significant differences from [BIK+92, PBI] in the structure of the argument. Omitted details are given in the full paper [BP93b].

We note that, independent of this work, Soren Riis (private communication) has shown similar results using methods of nonstandard model theory.

2 The Proof System $H + PHP_b(F)$

The proof system we use is based on the Frege proof system H from [BPU91, PBI, BIK+92] augmented with an axiom schema $PHP_b(F)$ which we describe below. H is a modification of a Frege system with NOT gates and unbounded fan-in OR gates. Excluded middle is the only axiom of H, and the cut rule is the main rule of inference, with extra merging and unmerging rules to manipulate the unbounded fan-in OR's. (By methods of Cook and Reckhow [CR77] the exact choice of the constant depth Frege system over unbounded fan-in $\wedge$, $\vee$, and $\neg$ we use is not crucial.) We now describe the axiom schema $PHP_b(F)$.

Let $F = \{F(i,j) \mid i \leq m+1, j \leq m\}$ be a set of bounded-depth formulas over the propositional variables $P_{\{i,j\}}$, $i,j \leq n$. The natural form of the pigeonhole principle using $\vee$ and $\neg$ based on F, $PHP(F)$, is an OR of the following subformulas: $C1(F,x) = \neg(F(x,1) \vee F(x,2) \vee ..., \vee F(x,n))$ for each $x \leq m+1$, which expresses the fact that x is not mapped to any hole; $C2(F,x_1,x_2,y) = \neg(\neg F(x_1,y) \vee \neg F(x_2,y))$ for each $x_1, x_2 \leq m+1$ and $y \leq m$, which expresses the fact that hole y has pigeons x_1 and x_2 mapped to it; and finally $C3(F,x,y_1,y_2) = \neg(\neg F(x,y_1) \vee \neg F(x,y_2))$ for each

$x \leq m+1$ and $y_1, y_2 \leq m$ which expresses the fact that pigeon x is mapped to holes y_1 and y_2:

Unfortunately $PHP(F)$ is not a convenient form of the pigeonhole principle for our purposes. For $x \leq m+1$ let x_i denote the i-th bit of x in binary notation. For each $F(x,y)$, we can express F in "left bitwise" notation by the formula $F_L(x,y)$:

$$F_L(x,y) = \neg \bigvee_{i=1}^{\lceil \log(m+1) \rceil} \neg F^L_{i,y_i}(x) = \bigwedge_{i=1}^{\lceil \log(m+1) \rceil} F^L_{i,y_i}(x),$$

where

$$F^L_{i,b}(x) = \bigvee_{z \leq m,\ z_i = b} F(x,z).$$

Similarly, we can also express F in the "right bitwise" notation by the formula $F_R(x,y)$:

$$F_R(x,y) = \neg \bigvee_{i=1}^{\lceil \log m \rceil} \neg F^R_{i,x_i}(y) = \bigwedge_{i=1}^{\lceil \log m \rceil} F^R_{i,x_i}(y),$$

where

$$F^R_{i,b}(y) = \bigvee_{z \leq m+1,\ z_i = b} F(z,y).$$

Let $PHP_b(F)$ denote the pigeonhole principle expressed as an OR of the following subformulas:

$$\{C1_b(F,x) \mid x \leq m+1\}$$
$$\cup \ \{C2(F_R, x_1, x_2, y) \mid x_1, x_2 \leq m+1,\ y \leq m\}$$
$$\cup \ \{C3(F_L, x, y_1, y_2) \mid y_1, y_2 \leq m,\ x \leq m+1\}\}$$

where

$$C1_b(F,x) = \neg \bigvee \{F^L_{i,b}(x) \mid 1 \leq i \leq \lceil \log m \rceil, b = 0,1\},$$

$$C2(F_R, x_1, x_2, y) = \neg(\neg F_R(x_1, y) \vee \neg F_R(x_2, y)),$$
$$C3(F_L, x, y_1, y_2) = \neg(\neg F_L(x, y_1) \vee \neg F_L(x, y_2))$$

Lemma 1: Let $F = \{F(x,y) \mid x \leq m+1, y \leq m\}$, where each $F(x,y)$ is a bounded-depth formula over the matching variables. Then there exists a bounded-depth, polynomial-size Frege proof of $PHP(F)$ from $PHP_b(F)$.

Let $F_b(x,*)$ denote the set of formulas $\{F_L(x,y) \mid y \leq m\}$. Similarly, let $F_b(*,x)$ denote the set of formulas $\{F_R(y,x) \mid y \leq m+1\}$. Note that for each x, all formulas in $F_b(x,*) \cup F_b(*,x)$ are boolean formula over the $O(\log m)$ subformulas, $\{F^L_{i,b}(x) \mid i \leq \lceil \log m \rceil,\ b \in (0,1)\}$, and $\{F^R_{i,b}(x) \mid i \leq \lceil \log(m+1) \rceil,\ b \in (0,1)\}$.

3 Restrictions, Decision Trees and the Switching Lemma

We now give the definitions for partial matching restrictions useful for the matching principle. These are entirely analogous to those in [BIK+92, PBI] but we include them for definiteness. The *variables over* D are $\{P_{\{i,j\}} : i \neq j \in D\}$. The i and j will be called the *endpoints* of $P_{\{i,j\}}$. For convenience we will write both P_{ij} and P_{ji} to represent variable $P_{\{i,j\}}$. A *map over* D is defined to be a conjunction of the form $\bigwedge \Gamma$, where Γ is a set of variables over D such that distinct variables in Γ have distinct endpoints. Maps describe partial matchings on the set D. The *size* of a map $\bigwedge \Gamma$ is $|\Gamma|$. An OR of maps is called a *map disjunction*. The *mapsize* of a map disjunction is the size of the largest map in the disjunction; if all the maps are of size at most t, then it will be called a *t-disjunction*. A map σ' *extends* map σ if $\sigma = \bigwedge \Gamma$ and $\sigma' = \bigwedge \Gamma'$ such that $\Gamma \subseteq \Gamma'$. We say that two maps σ and τ are *compatible* if there is some map π that extends both σ and τ and we denote the smallest such map π by $\sigma\tau$. A *truth assignment* φ over D is any total assignment of $\{0,1\}$ to the variables over D. Let $D' \subseteq D$. A truth assignment φ over D is a *matching on* D' if for all $i \in D'$ there is a unique $j \in D$ such that $P_{ij} = 1$.

If Y is a map or a set of variables, then $v(Y)$ denotes the set of endpoints of variables in Y.

We will now define a probability space of partial matchings on D, where $|D| = 2n+1$ The probability space $\mathcal{M}^D_p$ is the set of all pairs $\rho = < \pi, \pi_* >$ where π is a random matching of n edges in D and π_* is a random subset of the edges of π where each edge of π_* is chosen independently at random with probability p.

Every $\rho = \langle \pi, \pi_* \rangle$ in $\mathcal{M}^D_p$ determines a unique *restriction*, r, of the variables over D as follows.

$$r(P_{ij}) = \begin{cases} 1 & \text{if } \{i,j\} \in \pi \setminus \pi_* \\ 0 & \text{if there is an } k \text{ such that } \{i,k\} \\ & \quad \text{or } \{j,k\} \in \pi \setminus \pi_* \\ * & \text{otherwise} \end{cases}$$

In this way, the distribution $\mathcal{M}^D_p$ defines a probability distribution of restrictions. If r is a random restriction obtained by choosing a random ρ

according to $\mathcal{M}_p^D$, we will refer to both the restriction and the random partial matching by ρ. For a Boolean formula F and an element $\rho \in \mathcal{M}_p^D$, F restricted by ρ will be denoted by $F\restriction_\rho$.

3.1 Matching Decision Trees

A *matching decision tree* over domain D is defined as follows. It is a rooted tree where each interior node v is labelled by a query $i \in D$ and each edge is labelled by some pair $\{i, j\}$ where $j \neq i \in D$. Leaves are labelled with either "0" or "1". For each interior node v labelled by $i \in D$, there is exactly one out-edge labelled $\{i, j\}$ for each $j \in D \setminus \{i\}$ that does not appear in any edge label on the path from the root to v. The label of an interior node v may not appear in any edge label on the path from the root to v. Thus the set of edge labels on any path defines a map. A matching decision tree where all of the leaves are labelled "1" will be called a 1-*tree*. A matching decision tree T' *extends* a matching decision tree T if, for any root to leaf path p' in T', there is a unique path p in T such that the map σ' defined by p' extends the map σ defined by p. (Note that the leaf labels are not required to be related in this definition.)

A matching decision tree T over D *represents* a function f over domain D if for all leaf nodes $v \in T$, if we let σ be the map defined by the path in T from the root to v then for all truth assignments α over D that are matchings on $v(\sigma)$ and satisfy σ, $f(\alpha)$ is equal to the label of v. For a boolean function f over domain D, we define $d_D(f)$ to be the minimum height of all matching decision trees computing f.

Let T be a matching decision tree. In the remainder of this paper, the function represented by T is defined to be the map-disjunction, $maps(T)$, consisting of the labels of all of the paths in T that end in leaves labelled 1. Note that if T has height t, then the function computed by T is a t-disjunction. Furthermore note that for any partial matching restriction ρ over D, $maps(T\restriction_\rho) = maps(T)\restriction_\rho$.

Extending this definition, if f is a tree with intermediate nodes labelled by OR and NOT gates, and leaf nodes labelled by matching decision trees, then the function computed by f is obtained by iteratively computing the functions evaluated by the subtrees of f.

If ρ is a partial matching restriction over D and T is a matching decision tree over D, then define $T\restriction_\rho$ to be the decision tree obtained from T by removing all paths which have a label that has been set to "0" by ρ, and contracting all edges whose labels are set to "1" by ρ.

Lemma 2: Let f be a boolean function over D and let T be a matching decision tree representing f over D. If ρ is a partial matching restriction over D, then $T\restriction_\rho$ is a matching decision tree for $f\restriction_\rho$ over $D\restriction_\rho$.

Note that if T represents f over D then the tree T^c obtained by switching the 1's and 0's labelling the leaves of T represents $\neg f$. The lemma in the next section actually is a switching lemma in the spirit of [Hås87] because it will allow us to obtain a map disjunction that approximates the negation of f by representing f by a matching decision tree T and then taking $maps(T^c)$.

Where it is convenient, we shall assume that an ordering is given on D. Whenever we write a real number where an integer is required, we mean the integer part of the real number (floor). If f is a map disjunction defined over a set D and ρ is a restriction on D then we will use the notation $\delta(f\restriction_\rho)$ for $d_{D\restriction_\rho}(f\restriction_\rho)$. We now state the main combinatorial lemma.

Lemma 3 (Switching Lemma) Let f be an r-disjunction over $D \subseteq D^n$. Choose ρ at random from $\mathcal{M}_p^D$. If $s \geq 0$ and $pn \geq (r+s)(2r+2s+1)$ then

$$Pr[\delta(f\restriction_\rho) \geq s] < \alpha^s,$$

where $\alpha > 0$ satisfies $(1 + 225p^4n^3/\alpha^2)^r \leq 2$.

The inequality $(1+225p^4n^3/\alpha^2)^r \leq 2$ holds when $\alpha = 19p^2n^{3/2}r^{1/2}$. This can be seen by taking the natural logarithm of both sides and the applying the inequality $\ln(1+x) \leq x$.

The proof of the Switching Lemma is not included, but is similar to the proof in [PBI].

4 Exponential Lower Bounds

The overall structure of the exponential lower bound argument is very similar to the argument

in [KPW91] and in [BIK+92]. Given an alleged proof, P, of depth d and size S, a series of d restrictions are applied and after each one the proof is converted using a switching lemma to reduce the depth, until we end up with a sequence of formulas, each of which can be represented by matching decision trees. We will show that if P had size no greater than S, then after the d conversions, each matching decision tree is a 1-tree. But on the other hand, the final formula in the proof is the converted NPM formula, which becomes a matching decision tree which is not a 1-tree, and hence we have reached the contradiction.

The new part of the argument is showing that each instance of a PHP axiom schema gets converted into a 1-tree. This is the subject of section 5.

There are some formal and technical differences from [BIK+92, PBI] in how we apply the depth reduction in our argument. First of all, for convenience, we maintain a proof with small height matching decision trees at the leaves rather than formulas and finish reduction when each formula is a small height matching decision tree. More importantly, in order to preserve the formulas in the bit-wise version of the $PHP_b(F)$ axiom schema we do not always apply the switching lemma to $\vee$'s of decision trees. If the $\vee$ has fan-in at most $\log S$ then we simply 'stack' the decision trees one on the top of the other in the natural way creating a new deeper decision tree that evaluates all of the $\log S$ trees along each branch.

Using the switching lemma one can easily maintain via induction that after i levels of conversions have been applied the height of the decision trees at the leaves of the formulas in the proof is at most $\log^i S$. The domain, D, of the matching variables declines by a fixed fractional power at each step.

In this section we will prove the following theorem.

Theorem 4: Any proof of NPM_n in $H + PHP_b(F)$ of depth d must have size at least $S = \exp\left[\left(n^{6^{-(d+1)}}\right)\right]$.

Corollary 5: Any proof of NPM_n in $H + PHP_b(F)$ of polynomial-size must have depth $\Omega(\log\log n)$.

The Conversion Process

The conversion proceeds in rounds where each round reduces the depth of the formulas by 1. In each round subformulas lying just above the leaves are converted into decision trees. A certain set of these are converted using the switching lemma, others are converted by simpler means. Based on the set of subformulas for which the switching lemma is to be applied, a restriction σ is chosen to keep the heights of all the resulting decision trees small. Then the conversions themselves are done using the method previously decided upon for each subformula. A given subformula will, in general, appear several times throughout the proof. Each time it appears, the same conversion is applied.

More formally, after σ is applied, if f is $\neg T$ for some decision tree T then the conversion of f, $C[f] = T^c$ and if f is $\bigvee_{i=1}^{q} T_i$ then

(a) if $q > \log S$, then $C[f]$ is the result of applying the switching lemma argument to $\bigvee_{i=1}^{q} \mathit{maps}(T_i)$, and

(b) if $q \leq \log S$, then $C[f]$ is obtained by stacking the decision trees T_i such that a leaf at the end of path p is labelled "1" if p forces some T_i to 1. (One stacks decision trees T_1 and T_2 by replacing each leaf of T_1 by a copy of T_2, deleting incompatible paths, contracting redundant queries, and labelling the leaves of this copy of T_2 by the OR of the original leaf value and the the value of the leaf of T_1 that this copy of T_2 replaced.)

Note that f is $\neg T$ of f is $\bigvee_{i=1}^{q} T_i$ for $q \leq \log S$, i.e. the stacking method (b) is used to produce $C[f]$ then the application of any restriction ρ commutes with the conversion process on f, i.e. $C[f]\restriction_\rho = C[f\restriction_\rho]$, although this is not true in general if the switching lemma is used.

To prove Theorem 4, we will need the following theorem, which will be proven in the subsequent section.

Theorem 6: (PHP Axiom Soundness) Let $PHP_b(F)$ be an instance of the PHP_b axiom schema with depth at most d. Let T be the matching decision tree obtained by applying the conversion procedure d times to $PHP_b(F)$. Then T is a 1-tree.

Proof of Theorem 4

Let P be an alleged proof of NPM_n over D, $|D| = 2n+1$, of size less than S, and depth d (in $H + PHP_b(F)$). We will first show that there exists a sequence of good restrictions which allows us to convert the formulas in P into small-depth decision trees. Recall that each formula in P consists of d levels of OR's and NOT's, followed by the bottom level, which are depth-1 decision trees. Let $\rho^0, \rho^1, ..., \rho^d$ be a sequence of restrictions such that for all $0 \leq k \leq d$, ρ^k leaves all variables over D^{k+1} unset, $|D^k| = 2n_k + 1$. Let $P^1, ..., P^d$ be the sequence of formulas where P^k is equal to P^{k-1} converted by ρ^{k-1}. We will show that for all $i < d$, if each formula in P^i has depth $d-i$, mapsize t_i, and total size S, then P^i converted by ρ^i yields a new sequence of formulas, P^{i+1}, of depth $d-(i+1)$, mapsize t_{i+1} and total size S.

Let $t_0 = \log S$, and $t_i = t_0^i$ for $i > 0$. Define $\lambda(n) = n^{1/6}$, and $p_i = \lambda(n_i)/n_i$. If λ^i is the i-fold composition of λ with itself, then $\lambda^i(n) = n^{6^{-i}}$. If $S < \exp\left(n^{6^{-(d+1)}}\right)$, then for sufficiently large n, $t_0^d = (\log S)^d < \lambda^d(n)$.

Let D be the domain of the formulas in P^i. We can apply the Matching Switching Lemma, for ρ drawn at random from $\mathcal{M}_{p_i}^D$ to each distinct map disjunction in P^i. For each map disjunction, f, in P^i, for a randomly chosen $\rho \in \mathcal{P}_p^D$, the probability that $f\restriction_\rho$ cannot be represented by a matching decision tree over D^{i+1} of depth at most t_{i+1} is most $\alpha^{t_{i+1}}$, where $0 < \alpha < 19p_i^2 n_i^{3/2} t_i^{1/2}$. Because the size of P^i is at most S, there are at most S map disjunctions in P^i, and therefore, for a randomly chosen ρ, the probability that some map disjunction, $f\restriction_\rho$, in P^i cannot be represented by a depth-t_{i+1} matching decision tree over D^{i+1} is at most $S\alpha^{t_{i+1}} \leq S\alpha^{t_0} = S\alpha^{\log S}$. Since $p_i = \lambda(n_i)/n_i$,

$$\alpha < \frac{19\lambda(n_i)^2 t_i^{1/2}}{n_i^{1/2}} \leq \frac{\lambda(n_i)^{5/2}}{n_i^{1/2}} = \frac{n_i^{5/12}}{n_i^{1/2}} \leq \frac{1}{4}.$$

The second inequality holds because it can be shown that $t_i \leq n_i^{1/12}$. Therefore $S\alpha^{\log S}$ is at most $1/6$.

The expected number of stars (unset variables) after applying the restriction ρ is $n_i p_i = \lambda(n_i)$. Since the number of stars is binomially distributed, for sufficiently large n_0, a random ρ leaves at least the expected number of stars with probability greater than $1/3$. (See, for example, Lemma 4.1 of [BH]). Thus, there exists a restriction, ρ, leaving n_{i+1} stars, $n_{i+1} \geq \lambda(n_i)$, such that each decision tree in D^{i+1} has depth at most t_{i+1}. After applying this argument d times, we obtain a sequence $\mathcal{D}$ of decision trees over a smaller universe of size $2\lambda^d(n)+1$, where each tree has depth at most $t_d << 2\lambda^d(n)+1$.

Secondly, we will show that each decision tree in $\mathcal{D}$ is a 1-tree. The proof proceeds by induction on the number of trees in $\mathcal{D}$, or equivalently on the number of formulas in P. The base case is when P is a single formula. Therefore, it is either an instance of the PHP axiom or the excluded middle axiom. By Theorem 6, any instance of the PHP axiom is converted into a 1-tree. Otherwise, it is some excluded middle axiom, say $A \vee \neg A$. Let T_A be the decision tree that represents A. By definition, the decision tree representing $\neg A$ is the tree T_A, but with the opposite leaf labelling. Therefore, the decision tree for $A \vee \neg A$ is a 1-tree. For the inductive step, there are four cases, depending on the rule of inference. The more difficult cases are those involving unbounded fan-in OR gates—ie. the merging and unmerging rules. We will first sketch the proof when the inference is an application of the cut rule, and then when the inference is an application of unmerging.

Suppose that the inference is the cut-rule, and let A be the formula $X \vee Y$, let B be the formula $\neg X \vee Z$, and let C be the formula $Y \vee Z$. Let T_A, T_B and T_C be the corresponding decision trees in $\mathcal{D}$. We want to show that if T_A and T_B are 1-trees, then so is T_C. Let T_X, T_Y, T_Z be the decision trees that are obtained for the subformulas X, Y and Z, respectively during the conversion process. By definition of the conversion process, T_A is obtained by stacking the decision trees T_X and T_Y. Similarly, T_B is obtained by stacking the decision trees T_X^c and T_Z, and T_C is obtained by stacking the decision trees T_Y and T_Z. Suppose, for sake of contradiction, that a path, π, of T_C has leaf label 0. Thus there are compatible subpaths π_Y in T_Y and π_Z in T_Z that both have leaf labels 0. Since both T_C and T_X have height much smaller than the

universe size there is some path ρ in T_X (and thus also in T_X^c) compatible with π. By construction of T_A, $\sigma = \rho\pi_Y$ labels some path in T_A and by construction of T_B, $\tau = \rho\pi_Z$ labels some path in T_B. Now, in either T_X or T_X^c, the path ρ has leaf label 0. Thus either σ in T_A or τ in T_B has leaf label 0, a contradiction.

Intuitively, the above argument holds because, in the case of the cut rule, the OR gate and the negations involved in the inference are not approximated and therefore, since both antecedent formulas are 1, the derived formula should also be 1. Now consider the unmerging rule. Let A be the formula $\bigvee\{X1,..,Xn,Y1,..,Ym\}$, and let B be the formula $\bigvee\{\bigvee\{X1,..,Xn\},\bigvee\{Y1,...,Ym\}\}$, where B follows from A by the unmerging rule. Let T_A and T_B be the corresponding decision trees that are obtained by applying the conversion procedure. Assume that T_A is a 1-tree. Let $A' = \bigvee\{T_{X1},..,T_{Xn},T_{Y1},..,T_{Ym}\}$ denote the formula obtained during the conversion process, just before the final application of the switching lemma to obtain T_A. (Each T_{Xi} is a decision tree representing Xi.) Let T_X denote the decision tree after applying the switching lemma to $\bigvee\{T_{X1},..,T_{Xn}\}$, and define T_Y similarly. Then T_B will be obtained by stacking T_X and T_Y.

Fix a path, π labelling T_B. We want to show that π has leaf label 1. Since both T_A and T_B are decision trees of height much smaller than the universe size, there is some path ρ in T_A compatible with π. By assumption, the path ρ in T_A has leaf label 1. Since T_A represents A', it follows that $A'(\rho) = 1$. Therefore, $\bigvee\{\bigvee\{T_{X1},..,T_{Xn}\},\bigvee\{T_{Y1},..,T_{Ym}\}\}(\rho) = 1$. Thus either $\bigvee\{T_{X1},..,T_{Xn}\}(\rho) = 1$ or $\bigvee\{T_{Y1},..,T_{Ym}\}(\rho) = 1$, say the former. Since ρ is compatible with the path π_X in T_X that is a subpath of π, and T_X represents $\bigvee\{T_{X1},..,T_{Xn}\}$, this path π_X of T_X also has leaf label 1. Finally, because T_B is obtained by stacking the decision trees T_X and T_Y, the path of T_B labelled by π must have leaf label 1. The above argument shows that there exists a sequence of d restrictions such that P converts to a sequence of 1-trees. But this contradicts the fact that the final formula of P, NPM_n, converts to a 0-tree. □

5 The soundness of PHP_b

In this section we will prove Theorem 6. Since the decision trees produced are all of small height, in order to show that the tree produced by converting a pigeonhole axiom is a 1-tree, it suffices to show that it is impossible to force this tree to 0 by a small partial matching restriction.

Proof of Theorem 6. Consider an instance of the PHP_b axiom schema, $PHP_b(F)$, in the original proof, P, and let T be the tree into which $PHP_b(F)$ is converted by the conversion procedure. We wish to show that T is a 1-tree.

Suppose that T is not a 1-tree and thus it has a leaf labelled 0. Since the height of T is at most t_d there is some map σ of size $\leq t_d$ such that $T\restriction_\sigma = 0$.

Let $\pi_1,\ldots,\pi_d$ be the sequence of restrictions used in the conversion argument applied to proof P and let $\pi = \pi_1\pi_2\cdots\pi_d$ be their composition. For each proper subformula G of $PHP_b(F)$ consider the decision tree $T(G)$ created from G when the conversion process of the lower bound argument reaches the root node of G. We will use the notation G' to denote $T(G)\restriction_{\pi\sigma}$. Also, for any subformula G of $PHP_b(F)$ let $\pi(G)$ denote the portion of π that has been applied at the time that G is first converted to a decision tree.

By definition, $T\restriction_\sigma = 0$ if and only if $PHP_b(F)'$ is identically 0. The argument that the latter holds is based on the properties of the decision trees obtained for the various subformulas of $PHP_b(F)$. The easy cases are when some single decision tree expresses the fact that either the function is undefined on one of the $m+1$ points or that the function is not 1-1. The difficult case is when these decision trees do not obviously contradict the pigeonhole principle. That is, each one appears to define a part of a one-to-one function from $m+1$ to m.

If there were some partial matching, α, that extends some path in every tree, then it is easy to see that in this case $PHP_b(F)'$ is not identically 0 and we are finished. Unfortunately, this ideal situation may not hold because a particular partial matching may not extend any path in a given decision tree. For example if the root node of a tree queries i, then all matchings where i is unmatched will not extend any path of the tree. However, since we have required that the matching decision trees

are not too deep, we will still be able to show that $PHP_b(F)$ cannot be identically 0.

Theorem 7: $PHP_b(F)'$ is not identically 0.

Let the size of the universe remaining after $\pi\sigma$ is applied be $2n'+1$ and call the resulting domain D'. By construction, the tree $T(PHP_b(F))$ represents the $\bigvee$ of the maps in the various trees $T(C1_b(F,x))\restriction_{\pi'}$, $T(C2(F_R,x_1,x_2,y))\restriction_{\pi'}$ and $T(C3(F_L,x,y_1,y_2))\restriction_{\pi'}$ over $D\restriction_{\pi'}$ where $\pi' = \pi(PHP_b(F))$. We can apply the restriction $\pi\sigma$ to all of these trees and conclude that $PHP_b(F)'$ represents the $\vee$ of the maps in the various $C1_b(F,x)'$, $C2(F_R,x_1,x_2,y)'$, and $C3(F_L,x,y_1,y_2)'$ over D'. Thus it suffices to show that at least one of these trees has a branch with leaf label 1.

The trees $C1_b'$, $C2'$, and $C3'$ are obtained by applying the conversion argument working upwards from the trees $T(F(x,y))$ that represent the various $F(x,y)$, through the trees $T(F^L_{i,b}(x))$ and $T(F^R_{i,b}(y))$ for $x \leq m+1$ and $y \leq m$, to obtain the trees $T(C1_b)$, $T(C2)$, and $T(C3)$ which are then restricted by $\pi\sigma$. Since $\lceil\log(m+1)\rceil \leq \log S$, the conversion procedure will use the stacking method (b) at all OR's to produce the decision trees $T(C1_b)$, $T(C2)$, and $T(C3)$ from the bitwise trees $T(F^L_{i,b}(x))$ and $T(F^R_{i,b}(y))$. This means that the application of any restriction commutes with the conversion so we can obtain the same trees $C1_b'$, $C2'$, and $C3'$ by starting with the trees $F'^L_{i,b}(x)$ and $F'^R_{i,b}(y)$ for $x \leq m+1$ and $y \leq m$ and then applying the conversions afterward. This is convenient because it permits us to discuss all the decision trees at the upper levels of $PHP_b(F)$ over the same universe D'.

By the definition of the conversion, $F_L(x,y)$ will become the tree, $F'_L(x,y)$, obtained by stacking the $\lceil\log m\rceil$ trees, $F'^L_{i,y_i}(x)$. Similarly $F'_R(x,y)$ will obtained by stacking the $\lceil\log(m+1)\rceil$ trees $F'^R_{i,x_j}(y)$. For each x, $1 \leq x \leq m+1$ we can define $L_x = \{F'^L_{i,b}(x) \mid i \leq \lceil\log m\rceil, b \in \{0,1\}\}$, and similarly each tree $F'_R(y,x)$, $y \leq m$ is the composition of $\lceil\log(m+1)\rceil$ trees from the set of $2\lceil\log(m+1)\rceil$ trees $R_x = \{F'^R_{i,b}(Y) \mid i \leq \lceil\log(m+1)\rceil, b \in \{0,1\}\}$.

Therefore, for each x, the tree obtained by stacking the trees in $L_x \cup R_x$ is an extension of all of the trees $F'_L(x,*)$, and $F'_R(*,x)$. We define $\mathcal{T}_x$ to be this single matching tree over D', $|D'| = 2n'+1$, which simultaneously extends all of the trees $F'_L(x,*)$ and $F'_R(*,x)$, with the further modification that all the root-leaf paths are extended to some fixed length $k << n'$. This is accomplished by adding queries of other matching variables to any paths that are too short. Note that $\mathcal{T}_x$ still extends all of the trees in $F'_R(*,x)$ and $F'_L(x,*)$. The leaves of $\mathcal{T}_x$ are labelled with pairs $\{x \to u_1, \ldots, x \to u_k, v_1 \to x, \ldots, v_l \to x)\}$ where the pair $x \to u_i$ is a label of some path, p if and only if $F'_L(x,u_i)\restriction_p = 1$. Similarly, the pair $v_j \to x$ is a label of p if and only if $F'_R(v_j,x)\restriction_p = 1$. Note that since $F_R(*,m+1)$ is not defined, no leaf label of $\mathcal{T}_{m+1}$ will contain a pair $u \to m+1$, for any $u \leq m+1$. We now let $\mathcal{T} = \{\mathcal{T}_x \mid x \leq m+1\}$.

Definition. $\mathcal{T}$ is a *local function* if and only if: $\forall x \leq m+1$, $\forall$ paths p of $\mathcal{T}_x$, there exists some $z \leq m$ such that the leaf label of p contains the pair $x \to z$. In other words, if the map defined by p is ρ then there exists $z \leq m$ such that $F'_L(x,z)\restriction_\rho = 1$.

Definition. $\mathcal{T}$ is *locally 1-1* if for all x and for all paths, p, in $\mathcal{T}_x$, the leaf associated with p has at most one label of the form $x \to z_1$ and at most one label of the form $z_2 \to x$.

Definition. $\mathcal{T}$ is *consistent* if for all $x, y \leq m+1$ and for all pairs of paths, p_x in $\mathcal{T}_x$ and p_y in $\mathcal{T}_y$, if the maps they define, ρ_x and ρ_y respectively, are compatible then $x \to y$ labels the leaf of p_x if and only if $x \to y$ labels the leaf of p_y.

We first show that if either $\mathcal{T}$ is not locally 1-1 or not a local function then $PHP_b(F)'$ is not identically 0. Then we will argue that one of these cases must be true. We do this by showing, using the way that $\mathcal{T}$ is constructed, that if $\mathcal{T}$ is locally 1-1 then it is also consistent and then showing that it is impossible for $\mathcal{T}$ to be consistent as well as both a local function and locally 1-1. This latter proof requires a combinatorial argument.

Lemma 8: If $\mathcal{T}$ is not locally 1-1, then $PHP_b(F)'$ is not identically 0.

Proof: Assume that $\mathcal{T}$ is not locally 1-1. Then there exists an $x \leq m+1$, and a path p in $\mathcal{T}_x$ such that leaf label associated with p contains either (1) $x \to z_1$ and $x \to z_2$, $z_1 \neq z_2$, or (2) $z_1 \to x$ and

$z_2 \rightarrow x$, $z_1 \neq z_2$. Consider the first case. Let ρ be the map defined by p. Since $\mathcal{T}_x$ extends both $F'_L(x, z_1)$ and $F'_L(x, z_2)$,

$$F'_L(x, z_1)\restriction_\rho = F'_L(x, z_2)\restriction_\rho = 1$$

and thus $C2(F_L, x, z_1, z_2)'\restriction_\rho = 1$ which means that $PHP_b(F)'$ is not identically 0. The second case is handled similarly. □

Lemma 9: If $\mathcal{T}$ is not a local function, then $PHP_b(F)'$ is not identically zero.

Proof: If $\mathcal{T}$ is not a local function, then for some $x \leq m+1$, there exists a path, p, of $\mathcal{T}_x$, whose leaf label does not contain $x \rightarrow y$, for any $y \leq m$. Let ρ be the map defined by p. Since $\mathcal{T}_x$ extends all $F'^L_{i,b}(x)$, for every $i \leq \lceil \log m \rceil$ and $b = 0$ or 1 we have $F'^L_{i,b}(x)\restriction_\rho = 0$ and thus $C1_b(F, x)'\restriction_\rho = 1$ so $PHP_b(F)'$ is not identically 0. □

Lemma 10: If $\mathcal{T}$ is locally 1-1 then $\mathcal{T}$ is consistent.

Proof: We will prove the contrapositive. Suppose that $\mathcal{T}$ is not consistent. Then there exists $x, y \leq m+1$ and compatible maps, ρ_x labelling path p_x in $\mathcal{T}_x$, and ρ_y labelling path p_y in $\mathcal{T}_y$, such that either $F'_L(x, y)\restriction_{\rho_x} = 1$ and $F'_R(x, y)\restriction_{\rho_y} = 0$ or vice versa. We'll assume that the former case occurs (in which case we also know that $y \leq m$); the latter case is completely analogous.

We now sketch the remainder of the argument. Since $F_L(x, y)$ and $F_R(x, y)$ are constructed from the bit-wise versions of F, this inconsistency occurs exactly if x is mapped to at least two different z's in F, at least one of which agrees with y in each bit position (in effect the left bit-wise version sees a phantom edge not really present in F.) Thus the underlying F is not 1-1 and this is easily translated upward to show that $\mathcal{T}$ is not locally 1-1. The formal argument follows.

Recall that

$$F'_L(x, y)\restriction_{\rho_x} = C[\neg \bigvee_{i=1,..,\lceil \log m \rceil} \neg F'^L_{i,y_i}(x)]\restriction_{\rho_x}.$$

Because of the rules for conversion, the switching lemma is not used in producing $F'_L(x, y)$ from the various $F'^L_{i,y_i}(x)$ and so $F'_L(x, y)\restriction_{\rho_x} = 1$ implies that for all $i \leq \lceil \log m \rceil$, $F'^L_{i,y_i}(x)\restriction_{\rho_x} = 1$. By similar reasoning, $F'_R(x, y)\restriction_{\rho_y} = 0$ implies that there exists a j such that $F'^R_{j,x_j}(y)\restriction_{\rho_y} = 0$.

Now $F'^L_{i,y_i}(x) = T(F^L_{i,y_i}(x))\restriction_{\pi\sigma}$ and by definition $T(F^L_{i,y_i}(x))$ represents

$$\bigvee_{z \leq m, z_i = y_i} maps(T(F(x, z))\restriction_{\pi'})$$

over $D\restriction_{\pi'}$ where $\pi' = \pi(F^L_{i,y_i}(x))$ and thus $F'^L_{i,y_i}(x)$ represents

$$A^L_{i,y_i} = \bigvee_{z \leq m, z_i = y_i} maps(F'(x, z))$$

over D'. Similarly, $F'^R_{j,x_j}(y)$ represents

$$\bigvee_{w \leq m+1, w_j = x_j} maps(F'(w, y))$$

over D'.

Since $F'^R_{j,x_j}(y)\restriction_{\rho_y} = 0$ this implies in particular that $F'(x, y)\restriction_{\rho_y} = 0$. Also, since for each i, $F'^L_{i,y_i}(x)\restriction_{\rho_x} = 1$ this representation implies that there must be $z^1, ..., z^{\lceil \log m \rceil} \leq m$ such that for each i, $z^i_i = y_i$ and $F'(x, z^i)\restriction_{\rho_x} = 1$. Now, because ρ_x and ρ_y are compatible, $F'(x, y)\restriction_{\rho_y} = 0$ implies that $F'(x, y)\restriction_{\rho_x} \neq 1$. Thus for each i, $y \neq z^i$, and so there must be at least two different values $u \neq v$ among the z^i such that

$$F'(x, u)\restriction_{\rho_x} = F'(x, v)\restriction_{\rho_x} = 1.$$

We will now use this to show that $\mathcal{T}$ is not locally 1-1. Since $F'(x, v)\restriction_{\rho_x} = F'(x, u)\restriction_{\rho_x} = 1$, we have for each $i \leq \lceil \log m \rceil$,

$$A^L_{i,u_i}(x)\restriction_{\rho_x} = \left(\bigvee_{z \leq m, z_i = u_i} maps(F'(x, z)) \right)\restriction_{\rho_x} = 1$$

and

$$A^L_{i,v_i}(x)\restriction_{\rho_x} = \left(\bigvee_{z \leq m, z_i = v_i} maps(F'(x, z)) \right)\restriction_{\rho_x} = 1.$$

Since $\mathcal{T}_x$ extends every $F'^L_{i,b}(x)$, ρ_x fixes the value of every $F'^L_{i,b}(x)$, in particular of every $F'^L_{i,u_i}(x)$ and $F'^L_{i,v_i}(x)$. Because $F'^L_{i,u_i}(x)$ represents $A^L_{i,u_i}(x)$

over D', and $F'^{L}_{i,v_i}(x)$ represents $A^{L}_{i,v_i}(x)$ over D', for every i $F'^{L}_{i,u_i}(x) \restriction_{\rho_x} = A^{L}_{i,u_i}(x) \restriction_{\rho_x} = 1$ and $F'^{L}_{i,v_i}(x) \restriction_{\rho_x} = A^{L}_{i,v_i}(x) \restriction_{\rho_x} = 1$. Therefore, by construction, $F'_{L}(x,u)\restriction_{\rho_x} = 1$ and $F'_{L}(x,v)\restriction_{\rho_x} = 1$ and thus $x \rightarrow u$ and $x \rightarrow v$ both label the leaf followed by ρ_x in $\mathcal{T}_x$. Since $u \neq v$ this shows that $\mathcal{T}$ is not locally 1-1. □

Lemma 11: Let $\{\mathcal{T}_x \mid 1 \leq x \leq m+1\}$ be matching decision trees, as described above. Then it is impossible for $\mathcal{T}$ to be at the same time a local function, locally 1-1, and consistent.

Proof: Assume for sake of contradiction that $\mathcal{T}$ is locally 1-1, consistent, and a local function. By definition of $\mathcal{T}$, we also know that no leaf label of $\mathcal{T}_{m+1}$ contains $(x, m+1)$, for any x, $1 \leq x \leq m+1$. We will show that this leads to a contradiction.

Let U, V be maps over D of size exactly k. Let $\mathcal{T}_x$, $\mathcal{T}_y$ be complete matching decision trees over D. Then we have the following definitions.

(1) $r(U) = r_L(U) - r_R(U)$, where $r_L(U) = \#\{(x,y) \mid U$ labels a path in $\mathcal{T}_x$ mapping x to $y\}$, and $r_R(U) = \#\{(x,y) \mid U$ labels a path in $\mathcal{T}_y$ mapping x to $y\}$.

(2) $d(U,V) = \#\{(x,y) \mid U$ labels a branch in $\mathcal{T}_x$ mapping x to y and V labels a branch in $\mathcal{T}_y$ mapping x to y and U is compatible with V.

(3) Let $a(N,k)$ be the number of leaves in a complete, matching decision tree of height k over D, $|D| = N$.

(4) Let $b(N,r,k)$ be the number of leaves in a complete matching decision tree of height k over D, $|D| = N$, that lie below a given node of height r.

We will write $r_L(U)$ as $\sum_{(x,y)} r_L(U,x,y)$, where $r_L(U,x,y) = 1$ if U labels a path in $\mathcal{T}_x$ with leaf value (x,y), and otherwise $r_L(U,x,y) = 0$. Analogously, $r_R(U) = \sum_{(x,y)} r_R(U,x,y)$, where $r_R(U,x,y) = 1$ if U labels a path in $\mathcal{T}_y$ with leaf value (x,y). Similarly, we will write $d(U,V)$ as $\sum_{x,y} d(U,V,x,y)$, where $d(U,V,x,y)$ is 1 if: U is compatible with V; U labels a path in $\mathcal{T}_x$ with leaf value (x,y); and V labels a path in $\mathcal{T}_y$ with leaf value (x,y).

Lemma 12: Given the quantities defined above,

(a) $a(N-2k,k) \cdot r_L(U) = \sum_V d(U,V) \cdot b(N-2k, |U \cap V|, k)$.

(b) $a(N-2k,k) \cdot r_R(U) = \sum_V d(V,U) \cdot b(N-2k, |U \cap V|, k)$.

Proof: We give the proof of part (a). The proof of part (b) is analogous. Re-writing the left and right hand sides, we want to show:

$$\sum_{x,y} r_L(U,x,y) \cdot a(N-2k,k) = \sum_{x,y} \sum_{V} d(U,V,x,y) \cdot b(N-2k, |U \cap V|, k).$$

Fix U, x, y. Then we will show that $r_L(U,x,y) \cdot a(N-2k,k) = \sum_V d(U,V,x,y) \cdot b(N-2k, |U \cap V|, k)$. If $r_L(U,x,y) = 0$, then $d(U,V,x,y) = 0$ for all V, and therefore the above equality holds for these choices of U, x, y.

The other case is when $r_L(U,x,y) = 1$. Recall that U labels a path of $\mathcal{T}_x$ with leaf label $x \rightarrow y$ if and only if $r_L(U,x,y) = 1$. Let $\mathcal{T}' = \mathcal{T}_y \restriction_U$. We claim that the number of paths in $\mathcal{T}'$ equals $\sum_V d(U,V,x,y)$. To see that each path of $\mathcal{T}'$ contributes 1 to the quantity $\sum_V d(U,V,x,y)$, notice that if p is a path of $\mathcal{T}'$ labelled by V', then V' is compatible with U, and can be extended to a map, V, which labels a path of $\mathcal{T}_y$. Because the decision trees are consistent, since there is a path in $\mathcal{T}_x$ consistent with V and with leaf label $x \rightarrow y$, it must be the case that $x \rightarrow y$ is also a leaf label of the path labelled by V in $\mathcal{T}_y$, and therefore $d(U,V,x,y) = 1$. In the other direction, if $d(U,V,x,y) = 1$, then V is consistent with U, and V labels a path of $\mathcal{T}_y$ with leaf value $x \rightarrow y$, and therefore the restricted path, labelled by $V\restriction_U$, will be a path of $\mathcal{T}'$.

Let $\mathcal{T}''$ be the extension of $\mathcal{T}'$ to a complete, depth-k decision tree over D', $|D'| = N-2k$. Then the number of branches in the new, extended tree is exactly $\sum_V d(U,V,x,y) \cdot b(N-2k, |U \cap V|, k)$. Alternatively, the number of branches in $\mathcal{T}''$ is $a(N-2k,k)$, which is equal to $a(N-2k,k) \cdot r_L(U,x,y)$, and thus the lemma holds. □

We are now ready to complete the proof of Lemma 11. Recall that the decision trees $\mathcal{T}$ are

over the universe, D' of size $2n'+1$. Let $N = 2n'+1$. By the definition of $\mathcal{T}$, we know that for every U that labels a path in $\mathcal{T}_{m+1}$, there is no z such that the leaf label of U contains $z \to m+1$. Therefore, $r(U) > 0$ for those U's that label paths in $\mathcal{T}_{m+1}$. Secondly, because we are assuming that $\mathcal{T}$ is both locally 1-1 and a local function, we have that $r(U) \geq 0$ for every U. Therefore, $\sum_U r(U) > 0$, and thus $\sum_U a(N-2k,k)r(U) > 0$ as well.
By Lemma 12, we have $\sum_U \sum_V b(N-2k, |U \cap V|, k)[d(U,V) - d(V,U)] > 0$. However,

$$\begin{aligned}
&\sum_U \sum_V b(N-2k, |U \cap V|, k) d(U,V) \\
&= \sum_V \sum_U b(N-2k, |V \cap U|, k) d(U,V) \\
&= \sum_U \sum_V b(N-2k, |U \cap V|, k) d(U,V)
\end{aligned}$$

The first equality follows by swapping the summations and using the commutativity of intersection, and the second equality follows by switching notations for U and V. But this contradicts the inequality above, and therefore the lemma holds. □

Proof of Theorem 7. By Lemmas 10 and 11, if $\mathcal{T}$ is locally 1-1 then it cannot also be a local function. Thus $\mathcal{T}$ is either not locally 1-1 or not a local function and so, by Lemmas 9 and 8, $PHP_b(F)'$ is not identically 0. □

Theorem 6 now follows as an immediate corollary.

Acknowledgements

The authors would like to thank Russell Impagliazzo for his helpful comments and encouragement.

References

[Ajt88] M. Ajtai. The complexity of the pigeonhole principle. In *29th Annual Symposium on Foundations of Computer Science*, pages 346–355, White Plains, NY, October 1988. IEEE.

[Ajt90] M. Ajtai. Parity and the pigeonhole principle. In *Feasible Mathematics*, pages 1–24. Birkhauser, 1990.

[BH89] P. Beame and J. Håstad. Optimal bounds for decision problems on the CRCW PRAM. *Journal of the ACM*, 36(3):643–670, July 1989.

[BIK+92] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, P. Pudlák, and A. Woods. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the Twenty Fourth Annual ACM Symposium on Theory of Computing*, pages 200–220, Victoria, B.C., Canada, May 1992.

[BP93a] P. Beame and T. Pitassi. On the complexity of the parity argument. Preliminary manuscript, March 1993.

[BP93b] P. Beame and T. Pitassi. An Exponential Separation between the Matching Principle and the Pigeonhole Principle Technical Report. University of Washington, April 1993.

[BPU91] S. Bellantoni, T. Pitassi, and A. Urquhart. Approximation and small depth Frege proofs. To appear in *SIAM Journal of Computing*, December 1992.

[CR77] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1977.

[Hås87] Johan Håstad. *Computational Limitations of Small-Depth Circuits.* ACM doctoral dissertation award, 1986. MIT Press, 1987.

[KPW91] J. Krajíček, P. Pudlák, and A. Woods. Exponential lower bounds to the size of bounded-depth Frege proofs of the pigeonhole principle. Manuscript, 1991.

[Pap91] C. H. Papadimitriou. On inefficient proofs of existence and complexity classes. In *Proceedings of the 4th Czechoslovakian Symposium on Combinatorics*, 1991.

[PBI] T. Pitassi, P. Beame, and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity.* Accepted.

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