A resolution lower bound for a principle capturing the hardness of depth-1 LK

Alan Skelley*
Dipartimento di Informatica
Università degli Studi di Roma "La Sapienza"
alan@cs.toronto.edu

Neil Thapen†
Mathematical Institute
Academy of Sciences of the Czech Republic
thapen@math.cas.cz

Abstract

We introduce principles $2\mathrm{VR}^2$ and $2\mathrm{VR}^1$ which imply reflection for, respectively, the CNFs and the narrow CNFs refutable in the depth-1 propositional LK system PK_1. We give a polynomial-size refutation of their negations $\overline{2\mathrm{VR}^2}$ and $\overline{2\mathrm{VR}^1}$ in the system PK_1 and show an exponential lower bound on the size of their resolution refutations. We conjecture that they have no small $\mathrm{Res}(\log)$ refutations; this would be the first such lower bound for a principle with bounded depth refutations. We also show that if any CNF with a small PK_1 refutation is exponentially hard for $\mathrm{Res}(\log)$ then $\overline{2\mathrm{VR}^2}$ is as well.

1. Introduction

The refutational system $\mathrm{Res}(\log)$ is a strengthening of the well-known resolution system in which small conjunctions of literals, as well as single literals, are allowed to appear in clauses. It was introduced in [6] as a stronger, more robust version of resolution, corresponding to the bounded arithmetic theory T_2^2 via the Paris-Wilkie translation of first-order proofs into propositional proofs.

A current frontier in propositional proof complexity is to prove a lower bound for Res(log) for some CNF principle which has short bounded depth LK proofs (for example it is known that there is no small Res(log) proof of the pigeonhole principle, since it has no small proofs of any bounded formula-depth [1]). Recent work has looked at lower bounds for the weak pigeonhole from $2n$ to n. It was shown in [11] that this requires exponential size to refute if we restrict our conjunctions to size $\sqrt{\log n / \log\log n}$, and this was improved in [10] to $\varepsilon \log n / \log\log n$. But it is known that this is refutable with quasipolynomial size in Res(log) [8].

A candidate for a principle that may be hard for Res(log) is the finite Ramsey theorem, proposed in [4, 6]. This was shown to be provable in bounded depth in [9]. Currently not even resolution lower bounds are known for it; treelike $\mathrm{Res}(\log)$ lower bounds were shown in [6].

Obvious candidates are the reflection principles for proof systems thought to be stronger than $\mathrm{Res}(\log)$. These state that a proof system is sound and in a sense capture the strength of the system; however, they are generally very complicated. The contribution of this paper is to propose two CNFs, $\overline{2\mathrm{VR}^1}$ and $\overline{2\mathrm{VR}^2}$, which are as hard as reflection principles for a proof system PK_1 (corresponding to an extension of resolution in which we allow conjunctions of unrestricted size) but which are simple enough that we are able to prove at least a resolution lower bound for them.

Our motivation in this work is to better understand the relative strengths of the theories T_2^i in the bounded arithmetic hierarchy[2]. Here T_2^i is basically induction for formulas from level i of the polynomial hierarchy; these theories have close connections with complexity theory[5, 3, 12]. If we consider relatived versions $T_2^i(\alpha)$ with an undefined function symbol α (analogous to an oracle) then it is known that these theories are strictly increasing in strength. It is expected that they all even differ in their $\Sigma_1^b(\alpha)$ consequences, but the best known separation of this form is between $T_2^1(\alpha)$ and $T_2^2(\alpha)$. Finding a $\Sigma_1^b(\alpha)$ formula not provable in $T_2^2(\alpha)$ but provable somewhere higher in the hierarchy is closely related to the problem of finding lower bounds for Res(log) as described above.

The idea for our principles $2\mathrm{VR}^1$ and $2\mathrm{VR}^2$ comes from the characterization of the Σ_1^b consequences of T_2^3 in [7].

*Supported in part by grant LC505 (Eduard Čech Center) and NSERC PDF-313650-2005; This work was partly carried out while the author was at the Mathematical Institute of the Academy of Sciences of the Czech Republic in Prague.

†Supported in part by grant AV0Z10190503 and by the Eduard Čech Center grant LC505

A better than quasipolynomial Res(log) lower bound for $\overline{\mathrm{2VR}^1}$ would imply a $\Sigma_1^b(\alpha)$ separation between $T_2^2(\alpha)$ and $T_2^3(\alpha)$; a similar lower bound for $\overline{\mathrm{2VR}^2}$ would imply a $\Sigma_2^b(\alpha)$ separation.

In the next section we define our proof systems; then we describe our propositional principles $\overline{\mathrm{2VR}^1}$ and $\overline{\mathrm{2VR}^2}$ and prove our PK_1 upper bound and resolution lower bound; in the last section we give some more evidence that these are hard for Res(log) by showing that the existence of small Res(log) refutations of these principles would mean that every CNF with a small PK_1 refutation has a small $\mathrm{Res(log)}$ refutation.

2. Proof systems

PK is the propositional fragment of Gentzen's sequent calculus LK. In this paper, we shall consider small subsystems of PK, in particular operating over *cedents* (i.e. sets of formulas and no sequent arrow) and so more properly called Tait calculi. We shall consider them mainly as refutation systems.

The proof systems PK_1 and $\mathrm{Res(log)}$ are generalizations of resolution. In [6], Krajíček defines proof systems $R(f)$ by extending the language and rules of resolution to operate on clauses containing conjunctions of literals, and then restricting the sizes of these conjunctions to $f(S)$ in proofs of length S. $\mathrm{Res(log)}$ is our name for $R(log)$ from that paper. The inference rules of $\mathrm{Res(log)}$ are **cut**:

$$\frac{C \vee (\bigwedge_k l_k) \qquad D \vee \neg l'_1 \vee ... \vee \neg l'_j}{C \vee D}$$

provided $l'_1, ..., l'_j$ are among the l_ks and $k \geq 1$, and **$\wedge$-introduction**:

$$\frac{C \vee \bigwedge_j l_j \qquad D \vee \bigwedge_i l'_i}{C \vee D \vee (\bigwedge_j l_j \wedge \bigwedge_i l'_i)}$$

PK_1 allows clauses containing conjunctions of arbitrary size, and has the following inference rules (which seem to be slightly weaker than $R(id)$): **1-resolution** and an alternative **$\wedge$-introduction**:

$$\frac{C \vee (\bigwedge_k l_k \wedge l') \quad D \vee \neg l'}{C \vee D}; \qquad \frac{C \vee \bigwedge_j l_j \quad C \vee l'}{C \vee (\bigwedge_j l_j \wedge l')}$$

Our choice of rules for PK_1 is from [7], where they were chosen to simplify search problems involving large proofs. Cut can be simulated by repeated applications of 1-resolution, while the stronger $\wedge$-introduction can be simulated by allowing free introduction of axioms the form $\{p, \neg p\}$ and applying the weaker $\wedge$-introduction rule; we will therefore consider PK_1 to be largely equivalent to $R(id)$.

3. CNF principles and upper and lower bounds

$\overline{\mathrm{2VR}^1}$ is the following CNF, with a size parameter a. It is based on the "2-verifiable recursion machines" of [7]. For clarity we write it using quantifiers rather than propositional connectives. All quantifiers/connectives range over $[0, a]$.

1. For all x and all $i > 0$: $\exists! x' \, F_{ixx'}$
2. For all x: $\exists! y \, G_{xy}$
3. For all x, y, z: $G_{xy} \to V_{0xyz}$
4. For all x, y' and all $i > 0$: $\exists! y \, H_{ixy'y}$
5. For all y: $\exists! z \, S_{yz}$
6. For all y, z: $S_{yz} \to \neg V_{a0yz}$
7. For all x, y, z and all $i > 0$: $\exists! z' \, T_{ixyzz'}$
8. For all x, y, z, x', y', z' and all $i > 0$:

$$F_{ixx'} \wedge H_{ixy'y} \wedge T_{ixyzz'} \wedge \neg V_{ixyz} \to \neg V_{(i-1)x'y'z'}.$$

We think of the propositional variables G and S as representing the graphs of functions. F_i, H_{ix} and T_{ixy} represent the graphs of parametrized functions and V represents a parametrized relation $V_{ixy}(z)$.

The principle is best understood by considering a picture:

$$\begin{array}{ccccccccc}
V_a & & 0 & & & & y_a & \to \; S \to & z_a \\
 & & \vdots & & & & \vdots & & \vdots \\
V_i & & x & & & & y & & z \\
 & F_i & \downarrow & & & H_{ix} & \uparrow & T_{ixy} & \downarrow \\
V_{i-1} & & x' & & & & y' & & z' \\
 & & \vdots & & & & \vdots & & \vdots \\
V_0 & & x_0 & \to & G \to & & y_0 & & z_0
\end{array}$$

At the bottom, for $i = 0$, by axiom 3 we have $\forall x \, \exists y \, \forall z \, V_{0xy}(z)$ with y witnessed by G. Then by induction on i from the bottom up, for all i we can derive $\forall x \, \exists y \, \forall z \, V_{ixy}(z)$; axiom 8 gives us the induction step. But at the top, for $i = a$, by axiom 6 we have $\forall y \, \exists z \, \neg V_{a0y}(z)$ with z witnessed by S. This is a contradiction. This argument can be formalized into a polynomial-sized PK_1 refutation of $\overline{\mathrm{2VR}^1}$ which we will describe in a moment.

First we define another CNF, $\overline{\mathrm{2VR}^2}$, again with size parameter a:

1. For all x and all $i > 0$: $\exists! x'\, F_{ixx'}$

2. For all x: $\exists! y\, G_{xy}$

3. For all x, y, z: $G_{xy} \rightarrow V_{0xyz}$

4. For all x, y' and all $i > 0$: $\exists! y\, H_{ixy'y}$

5. For all y, z: $\exists z\, \neg V_{a0yz}$

6. For all x, y, z, x', y' and all $i > 0$:

$$F_{ixx'} \wedge H_{ixy'y} \wedge \neg V_{ixyz} \rightarrow \exists z'\, \neg V_{(i-1)x'y'z'}.$$

$\overline{2\mathrm{VR}^2}$ is a strengthening of $\overline{2\mathrm{VR}^1}$, which we get by replacing S and T (which we can think of as Herbrand functions) with disjunctions.

Lemma 1 $\overline{2\mathrm{VR}^1}$ *and* $\overline{2\mathrm{VR}^2}$ *have polynomial-size* PK_1 *refutations.*

Proof It is enough to consider the stronger principle $\overline{2\mathrm{VR}^2}$.

By axiom 3 and several $\bigwedge$-introduction steps, for all x, y we have $G_{xy} \rightarrow \bigwedge_z V_{0xyz}$. From axiom 2 for all x we have $\bigvee_y G_{xy}$. Several resolution steps give $\bigvee_y \bigwedge_z V_{0xyz}$ for each x.

This is the base case of the first order induction described above. To illustrate the inductive step, we will fix some $x = b$ and derive $\bigvee_y \bigwedge_z V_{1byz}$.

By axiom 1 we have $\bigvee_{x'} F_{1bx'}$. Call this A. By axiom 4, for all y' we have $\bigvee_y H_{iby'y}$. Call this B.

Temporarily fix values b' and c' for x' and y'. Let $\Gamma_{b'c'}$ be the conjunction $\bigwedge_{z'} V_{0b'c'z'}$. If we assume $\Gamma_{b'c'}$ then by cuts with instances of axiom 6 we can derive $\neg F_{1bb'} \vee \neg H_{ibc'y} \vee V_{1byz}$ for each y and z; then by several $\bigwedge$-introduction steps we get $\neg F_{1bb'} \vee \neg H_{ibc'y} \vee \bigwedge_z V_{1byz}$ for each y; then by repeatedly resolving with B for all ys (for $y' = c'$) we get $\neg F_{1bb'} \vee \bigvee_y \bigwedge_z V_{1byz}$.

By the induction hypothesis (here just the base case) we have already derived $\bigvee_{y'} \Gamma_{b'y'}$. Applying the above derivation to each disjunct Γ in turn, we get $\bigvee_{y'}[\neg F_{1bb'} \vee \bigvee_y \bigwedge_z V_{1byz}]$ which is just $\neg F_{1bb'} \vee \bigvee_y \bigwedge_z V_{1byz}$. Finally for each b' we resolve this with A to get $\bigvee_y \bigwedge_z V_{1byz}$, as required.

We repeat this for every value of x at every level i. At the top level we derive $\bigvee_y \bigwedge_z V_{a0yz}$. We then repeatedly cut with axiom 5 to derive the empty clause. □

3.1. Lower bound

Our lower bound proof is based on a bottleneck counting argument similar to those used in lower bounds for the pigeonhole principle. We give a lower bound for $\overline{2\mathrm{VR}^1}$ and the bound for $\overline{2\mathrm{VR}^2}$ follows.

For some large a, suppose that Π is a resolution refutation of an instance of $\overline{2\mathrm{VR}^1}$ of size a.

We find it intuitive to think about games rather than proofs, so we will think of Π as a strategy for the Prover in a certain Prover-Adversary game. In the game the Prover starts with an empty term (that is, an empty conjunction of the things he knows). In each turn he either asks the Adversary for the value of a variable and adds the answer to his term (corresponding to a resolution step) or forgets a variable from his term (corresponding to a weakening step). We make Π into a strategy for the Prover in this game by thinking of it as a directed acyclic graph, replacing each clause with its negation and reversing the direction of the arrows.

We will define a distribution on partial assignments to the variables of $\overline{2\mathrm{VR}^1}$. We then show that, if Π is small, then with high probability a partial assignment ρ has several nice properties and choose one fixed ρ with these properties. Then in particular, if the Adversary's replies are consistent with ρ then the terms known by the Prover are always narrow in a certain sense. This narrowness allows the Adversary always to give answers which are consistent with $\overline{2\mathrm{VR}^1}$, and hence the Prover is not able to win the game using the strategy Π.

Our distribution is determined by three parameters, a probability p, a width w and a constraint size c. We need that $pcw > a^{1+\varepsilon}$ for some $\varepsilon > 0$ and that $c \geq 4pa$. It is enough to take p to be $a^{-9/10}$ and w and c to be $a/10$. We do not try to optimize the numbers used here.

Below "exponentially high probability" means $> 1 - 2^{-a^{\varepsilon}}$ for some $\varepsilon > 0$ and "polynomially high probability" means $> 1 - a^{-\varepsilon}$ for some $\varepsilon > 0$.

We define a random partial assignment ρ as follows.

1. For each pair (i, x) with $i > 0$, with probability p "select" (i, x). Then for each row $i > 0$, suppose that distinct pairs $(i, x_1), \ldots, (i, x_m)$ were selected on that row. Randomly choose distinct numbers $x'_1, \ldots, x'_m$ (on row $i - 1$) and use these to partially define F_i as a partial injection. That is, for each j set $F_{ix_jx'_j}$ to be true and $F_{ix_jx''}$ to be false for every $x'' \neq x'_j$.

2. Similarly select each triple (i, x, y') for $i > 0$ with probability p and randomly define H_{ix} to be a partial

injection from the selected y's on row $i-1$ to a set of ys on row i.

3. Select each four-tuple (i,x,y,z) for $i>0$ with probability p and randomly define T_{ixy} to be a partial function from the selected zs on row i to a set of z's on row $i-1$. Here we can choose the z's independently since we do not insist it is a permutation.

4. For each (i,x,y) (possibly with $i=0$) with probability p set every value of $V_{ixy}(z)$ to be true or false uniformly at random (with probability $1/2$) – so each (i,x,y) either has all or none of the variables $V_{ixy}(z)$ set.

We say that (the value of) $F_i(x)$ is set in a partial assignment if $F_i(x)=x'$ is true for one x' and false for all the rest. Similarly for the other functions. We say V_{ixy} is set if it has a truth-value for every z.

Lemma 2 *By the Chernoff bound, with exponentially high probability, in every row i, $F_i(x)$ is set for at most $2pa$ values of x. Similarly for each function H_{ix} and T_{ixy} and for V_{ixy}. We may also assume that, if V_{ixy} is set by ρ, then it is made true for between $1/3$ and $2/3$ of the values z.*

Hence we may assume these bounds on size in the following lemmas, since any ρ for which they do not hold comes from an exponentially small error set which will not make a difference to our calculations.

We now extend the restriction to set all variables G and S in a way which satisfies axioms 2, 3, 5 and 6 of $\overline{2\mathrm{VR}^1}$.

5. For each x choose y at random amongst the $\geq a-2p$ values of y for which V_{0xy} was not set. Set $G(x)=y$ and set V_{0xy} to be true for all z.

6. Let y be on row a. There are two cases.

   Case 1: V_{a0y} is set. Then choose any z such that $V_{a0y}(z)$. Set $S(y)=z$.

   Case 2: V_{a0y} is not set. Then choose any z such that $T_{axy}(z)$ is not set. Set $S(y)=z$, make $V_{a0y}(z)$ false and make V_{a0y} true elsewhere.

Lemma 3 *With exponentially high probability, for any x, for all but $3a^2p^2$ numbers x' the following sets are disjoint in ρ: the domain of H_{1x}; the set of y' for which $V_{0x'y'}$ is set. Note that this second set includes $G(x')$.*

Proof For fixed x and x', the probability that there is some y' in the intersection from the original definition of ρ is $\leq ap^2$. The probability that $G(x')$ is in the domain of H_{1x} is $p<ap^2$. Hence by the Chernoff bound we may assume that there are $\leq 3a^2p^2$ many x's for which a bad y' exists. $\square$

Lemma 4 *With exponentially high probability, for any x and any $i>1$, for all but $2a^2p^2$ numbers x' the following sets are disjoint in ρ: the domain of H_{ix}; the set of y' for which $V_{(i-1)x'y'}$ is set.*

Proof For any i, x' and y', ρ puts y' into each of these sets independently with probability p. So the probability that y' is in both of them is $<p^2$. So the probability that, for fixed x', there is any y' in an intersection is $<ap^2$. Hence for each fixed i and x, by the Chernoff bound we may assume that there are $<2a^2p^2$ numbers x' for which the sets are not disjoint. $\square$

Definition 5 *We say a term t constrains $F_i(x)$ if either the literal $F_i(x)=x'$ is in t for some x', or the literal $F_i(x)\neq x'$ is in t for at least c many different x's. We define constraining $H_{ix}(y')$ and $T_{ixy}(z)$ similarly. Furthermore we say that t constrains V_{ixy} if either $V_{ixy}(z)$ or $\neg V_{ixy}(z)$ is in t for at least one z.*

A term t is F-fat if $|\{(i,x) : t \text{ constrains } F_i(x)\}|>w$. H-fat, T-fat and V-fat are defined similarly, in terms of the number respectively of triples (i,x,y'), 4-tuples (i,x,y,z) and triples (i,x,y) that are constrained by t.

Lemma 6 *Any F-fat term t is falsified by ρ with exponentially high probability. Similarly for H, T and V.*

Proof A difficulty here is that in each row the values ρ gives to each $F_i(x)$ are not set independently, since we insisted in the assignment that they should form a partial permutation.

We consider each row separately. Suppose t constrains F_i for $x_1,\ldots,x_s$ in row i. Let P_j be the probability that some literal involving x_j is falsified given that no literal involving x_k was falsified for any $k<j$. The "given …" part will make no difference to our bound on P_j, but we formally consider conditional probabilities because these events are not independent.

If $F_i(x_j)=x'$ is in t for some x', then $P_j>p(a-2pa-1)/(a-2pa)$ since, if ρ assigns a value to $F_i(x_j)$, then this value was chosen not from all of the numbers $<a$ but from a possibly smaller set, although one still of size $>a-2pa$ (here we are appealing to Lemma 2).

Similarly if $F_i(x_j)\neq x'$ is in t for at least c many different x's, then $P_j>p(c-2pa)/(a-2pa)$.

Since $c\geq 4pa$, both these probabilities are $>pc/2a$. So the probability that no literal in row i is falsified is $<(1-$

$pc/2a)^s$ and, as the rows are independent, the probability that no literal in any row is falsified is $< (1 - pc/2a)^w \leq e^{-pcw/2a} = 2^{-a^{1/10}/200}$. $\square$

We now define an important tool in the Adversary's strategy. We will need this to describe one more nice property of ρ, which will hold with polynomially high probability.

Definition 7 *An F-path in a partial assignment α is a maximal sequence $x_j, x_{j-1}, \ldots, x_k$ with $j > k$ such that $F_j(x_j) = x_{j-1}, \ldots, F_{k+1}(x_{k+1}) = x_k$.*

Given a F-path σ of the above form, a H-path τ which "matches" σ is a maximal sequence $y_l, y_{l+1}, \ldots, y_m$ such that $k \leq l < m \leq j$ and $H_{(l+1)x_{l+1}}(y_l) = y_{l+1}, \ldots, H_{mx_m}(y_{m-1}) = y_m$. Notice that you cannot have an H-path without an F-path, and that an F-path may have several disjoint, overlapping matching H-paths.

We say that x_j is at the top of the F-path and x_k is at the bottom, and the length of the path is $j - k$. The domain of the path is $[k, j]$. Similarly for H-paths.

We say that such a path τ is "full" if for each $i \in [l, m]$, $V_{ix_iy_i}$ is set, and for each $i \in [l+1, m]$, $T_{ix_iy_i}$ is total, and that these hold in a way consistent with $\overline{2\mathrm{VR}^1}$.

Definition 8 *Let γ be a partial assignment (which will eventually represent the term remembered by the Prover at a round in the game). We say that a partial assignment α is a completion of γ if all of the following hold.*

1. *In α, where they are defined, F, H and T are partial functions for every choice of parameters, and furthermore F and H are partial permutations.*
2. *$\rho \subseteq \alpha$.*
3. *$\gamma \subseteq \alpha$.*
4. *Every $F_i(x)$ constrained in γ is set in α.*
5. *For every $H_{ix}(y)$ constrained in γ, $F_i(x)$ and $H_{ix}(y)$ are set in α.*
6. *Every $T_{ixy}(z)$ constrained in γ is set in α.*
7. *Every V_{ixy} constrained in γ is set in α.*
8. *Every H-path in α is full. We call this condition "fullness".*
9. *No variables are set in α other than those as required above.*
10. *If V_{ixy} is true for all z, then $i \leq 2w+2$ and the F-path containing (i, x), if there is one, has all of its domain $\leq 2w + 2$. We call this "uselessness".*
11. *No F-path contains more than two triples (i, x, x') such that $F_i(x) = x'$ was set in ρ (rather than coming from γ). We call this "avoiding paths from ρ".*
12. *α is consistent with $\overline{2\mathrm{VR}^1}$.*

Lemma 9 *With polynomially high probability there is a completion α_0 of the empty assignment.*

Proof We need to show how to extend ρ to make it full. The only obstacle to this is if there are some H-path on which the Vs and Ts were badly defined. Note that if we were proving a lower bound for $\overline{2\mathrm{VR}^2}$ rather than for $\overline{2\mathrm{VR}^1}$ then this lemma would be much shorter, since if we do not have the function T then the only obstacle to making an H-path full is if a V is set all-true somewhere on the path (which can only happen at the bottom, in ρ) and a V is set not-all-true somewhere higher on the path.

We may assume that there is no H-path in ρ of length two or more, by the following calculation. For any $i, x, x', x'', y, y', y''$, the probability that x, x' and x'' form an F-path and a matching H-path (at level i) is $(p/a)^4$. There are only a^7 such 7-tuples, so the probability that any of them form such a path is $< (p/a)^4 a^7 = p^4 a^3 = a^{4/10}/a$.

We can also show that there is no H-path of length one such that V is set at both the top and bottom of the path. For this there are three cases.

For any i, x, x', y, y' with $a > i > 1$ the probability that they form an F-path and a matching H-path (at level i) is $(p/a)^2$, and the probability that V_{ixy} and $V_{(i-1)x'y'}$ are both set is p^2. There are a^5 such 5-tuples, so the probability that any of them form such a bad configuration is $< (p/a)^2 p^2 a^5 = p^4 a^3 = a^{4/10}/a$.

For $i = a$, if $x \neq 0$ then the calculation is as above. If $x = 0$, then V_{a0y} is set for every y (by part 6 of the definition of ρ). But this is only a problem if there exist x', y', y such that $0, x$ is a F-path, y', y is a matching H-path and $V_{(a-1)x'y'}$ is set, which happens with probability $< a^{3/10}/a^2$.

For $i = 1$, the probability that $V_{(i-1)x'y'}$ is set is slightly higher than p, since $V_{0x'G(x')}$ is always set, but it is no more than $p + 1/(a - 2p)$ so the probability of a bad configuration is still polynomially small.

So ρ contains several H-paths of length 1, for which V may have been set by ρ at one end or the other, but not at both ends. Suppose y', y is such an H-path which matches a F-path x, x' (with x, y at level i and x', y' at level $i-1$). T_{ixy} is partially defined, with domain of size $< 2pa$. Suppose V_{ixy} was set by ρ. Then there is no difficulty in setting $V_{(i-1)x'y'}$ and extending T_{ixy} to a total function in such a way that (a) axiom 8 of $\overline{2\mathrm{VR}^1}$ is satisfied and (b) $V_{(i-1)x'y'}(z')$ is false for some z', so that we do not violate uselessness. The situation is similar if instead $V_{(i-1)x'y'}$ was set by ρ, or if neither was set.

For part 11 of the definition, "avoiding paths from ρ", we need to bound the probability that ρ contains any F-paths of length three or more. By a similar calculation to the above, this is $< (p/a)^3a^5 = p^3a^2 = a^{3/10}/a$. □

Notice that Vs only appear in $\alpha_0 \setminus \rho$ on H-paths in ρ, so we will still be able to use Lemma 4. See below.

Theorem 10 *For some $\varepsilon > 0$, there is no resolution refutation of $\overline{2\mathrm{VR}^1}$ of size $< 2^{a^\varepsilon}$.*

Proof Suppose that there is a subexponential-size proof Π. Then we can treat Π as a Prover-strategy, as outlined above, and by Lemma 6 we can find a single partial assignment ρ which falsifies every fat term in Π. Furthermore we can choose ρ to satisfy all our other lemmas.

The Prover begins the game with an empty assignment γ. At each turn, he can either forget a variable from γ or ask the Adversary for the value of a variable.

We have shown that, provided that the Adversary's replies are consistent with ρ, then the Prover's strategy remains "narrow" and γ constrains at most w many Fs, Hs, Ts and Vs.

The Adversary's strategy is to maintain a completion α of γ. As long as he does this, γ must be consistent with $\overline{2\mathrm{VR}^1}$ and the Prover cannot win.

The Adversary begins with α_0, which is by definition a completion of the empty assignment.

We consider a turn in the game where the Prover starts with an assignment γ and ends with an assignment γ'. The Adversary starts with a completion α of γ. We must show there is a completion α' of γ'.

If the Prover forgets a variable and so $\gamma' \subseteq \gamma$, then the adversary can easily find a completion $\alpha' \subseteq \alpha$.

Suppose the Prover asks "does $F_i(x) = x'$?". If $F_i(x)$ is set in α, then the Adversary answers appropriately and α is still a completion. If $F_i(x)$ is not set, then let X be the set of x''s for which $F_i(x) \neq x''$ is in γ. If $|X| < c-1$ (or $x' \in X$) then the Adversary replies "no" and adds $F_i(x) \neq x'$ to α. If $|X| = c-1$ and $x' \notin S$ then any answer to the Prover's question will inevitably constrain $F_i(x)$, so α' must set a value for $F_i(x)$ (and the Adversary replies accordingly). We show how to set a value for $F_i(x)$ in Lemma 11 below.

Suppose the Prover asks "does $H_{ix}(y') = y$?". This is similar to the case for F, but now we need to keep track of whether $H_{ix}(y')$ is constrained in γ even if $H_{ix}(y')$ is already set in ρ, because of part 5. of definition 8. So if $H_{ix}(y')$ is set in α, then the Adversary answers appropriately. When $H_{ix}(y')$ becomes constrained in γ, first the Adversary uses Lemma 11 to set $F_i(x)$ in α, and then uses Lemma 12 to set $H_{ix}(y')$ (if it was not set already).

T is dealt with the same way as F. When $T_{ixy}(z)$ becomes constrained in γ and is not already set in α, the Adversary gives it a value arbitrarily. A single query $V_{ixy}(z)$ is enough to constrain V_{ixy}. If it is not already set in α, the Adversary assigns it values arbitrarily, making at least one of them false to maintain uselessness. These steps could lead to a contradiction with $\overline{2\mathrm{VR}^1}$ if the Ts or Vs lie on some H-path; but by fullness all the Ts and Vs on H-paths are already set by α in a consistent way. □

Lemma 11 *There is a way to extend α to α' which sets a value for $F_i(x)$.*

Proof **Case 1: $i > 1$.**

We list the properties which we do not want our choice x' of a value for $F_i(x)$ to satisfy, and give bounds on the number of x's with each bad property.

1. $F_i(x) \neq x'$ is in γ; $\leq c$.
2. $F_i(\tilde{x}) = x'$ is in α for some $\tilde{x} \neq x$; $\leq 2pa + 2w$ ($2pa$ for Fs set by ρ, w for Fs constrained by γ and w for Hs constrained by γ – see part 5 of Definition 8).
3. $F_{i-1}(x')$ is set in α; $\leq 2pa + 2w$ (as above).
4. $V_{(i-1)x'y'}$ is set in γ for some y'; $\leq w$.
5. x' does not satisfy Lemma 4; $\leq 2a^2p^2$.

The sum of these bounds is less than a, so some good x' exists. We may extend α by setting $F_i(x) = x'$. Notice that by item 3, this does not grow any F-paths upwards, so uselessness is preserved; also by item 3 we continue to avoid paths from ρ.

We now need to add some more things to α to get fullness. By 2 and 3, x' is not on any F-path in α. Hence any $V_{(i-1)x'y'}$ that is set, was set either by ρ or in γ. By item 4, it can only have been set in ρ. By part 5 of definition 8, $H^\alpha_{ix} = H^\rho_{ix}$. Hence by Lemma 4, there is no y' such that $y' \in \mathrm{dom} H^\alpha_{ix}$ and $V_{(i-1)x'y'}$ is set in α.

So if y', y is (part of) any H-path matching our new (partial) F-path x, x', then there is no difficulty in setting $V_{(i-1)x'y'}$, setting V_{ixy} if necessary and extending T_{ixy} to a total function in a way consistent with $\overline{2\mathrm{VR}^1}$ and with uselessness, as in the proof of lemma 9.

Case 2: i=1.

We list the properties which we do not want our choice x' of a value for $F_1(x)$ to satisfy, and give bounds on the number of x's with each bad property.

1. $F_1(x) \neq x'$ is in γ; $\leq c$.
2. $F_1(\tilde{x}) = x'$ is in α for some $\tilde{x} \neq x$; $\leq 2pa + 2w$.
3. $V_{0x'y'}$ is set in γ for some y'; $\leq w$.
4. x' does not satisfy Lemma 3; $\leq 3a^2p^2$.

Fullness is preserved as in case 1.

For $y' = G(x')$, $V_{0x'y'}$ will be all true in ρ. But by the condition on avoiding paths from ρ and the limit on the number of Fs constrained by γ, the F-path containing $(0, x')$ has length at most $2w + 2$, so uselessness is preserved. □

Lemma 12 *There is a way to extend α to α' which sets a value for $H_{ix}(y')$.*

Proof Recall that this lemma is only applied once $F_i(x)$ is set to some value x'. Note that $F_i(x)$ is not necessarily constrained by γ, but other than this α is a completion of γ.

Case 1: $0 < i < a$, there is some x^* such that $F_{i+1}(x^*) = x$.

We list the properties which we do not want our choice y of a value for $H_{ix}(y')$ to satisfy, and give bounds on the number of ys with each bad property.

1. $H_{ix}(y') \neq y$ is in γ; $\leq c$.
2. V_{ixy} is set in ρ or constrained by γ; $\leq 2pa + w$.
3. $y \in \mathrm{ran}H^{\alpha}_{ix}$; $\leq 2pa + w$.
4. $y \in \mathrm{dom}H^{\alpha}_{(i+1)x^*}$; $\leq 2pa + w$.

The sum of these bounds is less than a, so some good y exists. We may extend α by setting $H_{ix}(y') = y$.

By items 3 and 4, (i, x, y) is not on any H-path in α, so together with item 2 this means that V_{ixy} is not set in α. Hence there is no difficulty with setting V_{ixy}, possibly setting $V_{(i-1)x'y'}$ and extending T_{ixy} to achieve fullness.

If $V_{(i-1)x'y'}$ is all true, then we must set V_{ixy} to be all true. But this does not contradict uselessness, since the F-path containing $(i-1, x)$ does not grow upwards.

Case 2: $0 < i < a$, there is no x^* such that $F_{i+1}(x^*) = x$.

The argument is as above, but simpler because we do not have to worry about $\mathrm{dom}H_{(i+1)x^*}$.

Case 3: $i = a$, $x = 0$.

Notice that by our construction of ρ for all y, V_{a0y} is set in ρ and $V_{a0y}(S(y))$ is false.

Again we list the ys to avoid:

1. $H_{a0}(y') \neq y$ is in γ; $\leq c$.
2. y is already in $\mathrm{ran}H^{\alpha}_{a0}$; $\leq 2pa + w$.
3. V_{a0y} was set at random in ρ, at step 4 in the definition of ρ; $\leq 2pa$.
4. V_{a0y} was set at step 6 in the definition of ρ, and $T_{a0y}(S(y))$ is constrained by γ; $\leq w$.

We may find a good y and extend α by setting $H_{a0}(y') = y$.

We may assume that $V_{(a-1)x'y'}$ is set and, by uselessness, that it is false for at least one z'. By item 3 above and case 2 of part 6 of the definition of ρ, $z = S(y)$ is the only false z in V_{a0y} and $T_{a0y}(z)$ was not set in ρ, so is not set yet in α. Set $T_{a0y}(z) = z'$. Then extend T_{a0y} arbitrarily to a total function. This gives fullness.

Case 4: $i = a$, $x \neq 0$.

This is the same as case 2. □

4. Reflection principles and propositional translations

In this section we show that if our CNF principles have small refutations in Res(log), then everything with a small refutation in the possibly-stronger system PK_1 already has a small refutation in Res(log). This is corollary 16 below.

This could be shown by directly constructing Res(log) refutations, but instead we will describe some first-order proofs in bounded arithmetic and make use of known translations of these into propositional proofs.

We begin by defining some first-order reflection principles for PK_1 and saying how these are related to our principles $\overline{2\mathrm{VR}^1}$ and $\overline{2\mathrm{VR}^2}$.

4.1. Reflection and verifiable recursion principles

A reflection principle is a formula expressing that, if a formula in a certain class is provable in a certain proof system, then that formula is true.

We are interested here in what might be called "second-order" reflection principles, and in ones which deal with refutations rather than proofs. We are given a size parameter a and some oracles Γ, Π and α. The principle states that if we interpret these oracles as structures on a domain of size a, then if Π is a refutation of the formula Γ, then Γ is false under the assignment α.

Reflection principles of this form are developed in [7], for the resolution and PK_1 proof systems. That paper goes into some detail about how to present the structure of a formula and a PK_1 refutation using polynomial time (possibly with an oracle) relations and functions. Π will contain machinery to list the literals in a narrow clause or conjunction, to find the hypotheses and rule used to derive a given clause, to check membership of a literal in a clause or a conjunction, etc. Narrow here means of size polynomial in $|a|$.

$1\mathrm{-Ref}(\mathrm{PK}_1)(\Gamma, \Pi, \alpha, a)$, with size parameter a, is a $\Sigma_1^b(\Gamma, \Pi, \alpha)$ formula stating that either the refutation Π of a set of narrow clauses Γ is ill-defined, or there is an initial clause falsified by α.

$2\mathrm{-Ref}(\mathrm{PK}_1)(\Gamma, \Pi, \alpha, a)$ is analogous but without restricting Γ to be narrow. Now the statement that a particular initial clause is falsified by α is Π_1^b, so $2\mathrm{-Ref}(\mathrm{PK}_1)(\Gamma, \Pi, \alpha, a)$ is $\Sigma_2^b(\Gamma, \Pi, \alpha)$.

A verifiable recursion program (defined in [7]) is a computational object consisting of a sequence of machines (visualized proceeding from top to bottom), each able to make recursive queries to other machines further down the list. The output of any of these machines is locally verifiable by a correctness predicate, and the output of the program on an input x is the output of the topmost machine on that input. (For simplicity, we will usually just use 0 as the top-level input to the program.) There are several variations of this model depending on the depth of the sequence as a function of the input size, the semantics for the recursive calls, the complexity of the correctness predicate, and so on.

For the purposes of this paper, we will focus on **2-verifiable recursion programs** of the following kind: (The 2 refers to the extra "check" argument to the correctness predicates). For size parameter a (for inputs in the range $[0, a]$) there are $a+1$ machines, from number a at the top to 0 at the bottom. The correctness predicate, V_i, at each level is ternary and takes a check argument in addition to the input and output. Each machine makes exactly one recursive call to the next lower machine, formalized by $F_i(x)$. Its output y on input x is a function of this return value y', formalized as $H_i(x, y')$, and is said to be correct if $\forall z V_i(x, y, z)$. At the bottom level (0), correct outputs can be found in polynomial time and we formalize this with a function G and assert that $\forall x, z V_0(x, G(x), z)$ holds for a well-defined program. At level $i > 0$ there is optionally a Herbrand function T to help enforce the correctness of the output of machine i: in a well-defined program this function has the property that $\forall i > 0 \, \forall x, y', z \, (\neg V_i(x, H_i(x, y'), z) \rightarrow \neg V_{i-1}(F_i(x), y', T_i(x, y, z)))$. A well-defined program without the Herbrand function need satisfy only the weaker property that $\forall i > 0 \, \forall x, y' \, (\exists z \neg V_i(x, H_i(x, y'), z) \rightarrow \exists z' \neg V_{i-1}(F_i(x), y', z'))$. Finally, at the top level a, there is optionally another Herbrand function S to produce the check argument, and in this case the correctness condition at the top is weakened to $V_a(x, y, S(y))$. For simplicity, we will use 0 as the input to the top level when asserting totality; this does not make the resulting principle any weaker.

With this description in mind, we define the first-order versions of our CNFs: $2\mathrm{VR}^1(V, F, G, H, S, T, a)$ is $\forall \Sigma_1^b$ and states that if a 2-verifiable recursion program with Herbrand functions is well-defined, then it is total:

$$\begin{aligned} &[\forall x, z V_0(x, G(x), z)] \wedge \\ &[\forall i > 0, x, y', z \, (V_{i-1}(F_i(x), y', T_i(x, y', z)) \\ &\qquad \rightarrow V_i(x, H_i(x, y'), z))] \rightarrow [\exists y V_a(0, y, S(y))] \end{aligned}$$

$2\mathrm{VR}^2$, meanwhile, omits the Herbrand functions S and T and is $\forall \Sigma_2^b$:

$$\begin{aligned} &[\forall x, z V_0(x, G(x), z)] \wedge \\ &[\forall i > 0, x, y' (\forall z' V_{i-1}(F_i(x), y', z') \rightarrow \\ &\qquad \forall z V_i(x, H_i(x, y'), z))] \rightarrow [\exists y \forall z V_a(0, y, z)] \end{aligned}$$

In each case, all quantifiers are bounded by a. The hypothesis of the implication we call *well-definedness* of the program and the conclusion *totality*. Our CNF $\overline{2\mathrm{VR}^1}$ is the propositional translation of the negation of $2\mathrm{VR}^1$ and similarly for $\overline{2\mathrm{VR}^2}$ and $2\mathrm{VR}^2$.

Theorem 13 *There is a term t and polynomial-time V, F, G, H, S, T defined in terms of oracles Γ, Π, α such that*

1. $T_2^2 \vdash 2\mathrm{VR}^1(t(a)) \rightarrow 1\mathrm{-Ref}(\mathrm{PK}_1)(a)$ *and*
2. $T_2^2 \vdash 2\mathrm{VR}^2(t(a)) \rightarrow 2\mathrm{-Ref}(\mathrm{PK}_1)(a)$.

Proof We begin with part 1. This is essentially proved in [7], but for a slightly more general kind of recursion program allowing recursive calls to an arbitrarily lower machine. We will present this construction at a high level, and point out how to modify it for the present theorem.

We are given Γ, Π, α with size parameter a. Say that Π consists of b clauses $C_0, ..., C_b$. The 2VR program we construct will have size parameter t, a term in a which we will not calculate here. The general outline of the program is that machine i (for $i \leq b$) will attempt to find a true literal or conjunction in C_i. Note that the lower-down machines of the program correspond to the upper (towards the initial) clauses of the proof. The correctness predicate $V_i(x, y, z)$ is true if either y really is a true literal, or if it names a conjunction and z is not a witness that the conjunction is actually false. (The input x is not used in the original construction). Machines i for $i > b$ behave as machine b. The Herbrand function $S(y)$ for the top level simply returns 0.

If C_i is an initial clause (and therefore narrow), machine i examines the clause and returns a true literal if one exists. Otherwise, depending on the PK_1 rule used to derive clause C_i and possibly the assignment to a relevant literal, machine i queries the machine for one of the hypotheses of this rule. The return value of machine i is then a simple function of this reply.

For some inference rules involving conjunctions, machine i may return a conjunction without being able to verify that it is satisfied (as it may contain too many literals to check); however in these cases, given a false literal in the conjunction, the Herbrand function T_i is able to pass the blame downwards to the machine for the relevant hypothesis.

Now, if a machine corresponding to an initial clause is not total, this yields a falsified initial clause. Every other way that the verifiable-recursion program could be ill-defined yields a witness to the PK_1 refutation being badly formed. Finally, a correct output at the top yields a literal or conjunction in the final clause of the refutation, which is another witness of it being ill-formed (as the final clause should be empty). All this is easily provable in T^2_2, as it involves only polynomial-time reasoning about the soundness of rules of PK_1

This construction is easily adapted to the present case: we must only arrange for the machine at each level to make its single recursive call to the next-lower level. This is done by expanding the inputs and outputs of the machines to allow a call from level i to reach level $j < i$ by being passed down (and back up) through the intermediate machines as a special case; similarly for the outputs and check-arguments. Now provably in T^2_2, all witnesses of ill-definedness of the new machine imply such a witness for the original machine.

The statement for $2\mathrm{VR}^2$ and $2{-}\mathrm{Ref}(\mathrm{PK}_1)$ is proved with the identical construction, except that the Herbrand function is omitted. Ill-definedness of the machine at the bottom yields a false initial clause (which now could be wide). Ill-definedness in the middle means that the refutation is badly formed (as T^2_2 still proves the soundness of individual rules of PK_1), and a correct output at the top again is a witness that the final clause is not empty. □

4.2. Propositional translations

We use a theorem of Krajíček, which is a model-theoretic version of the Paris-Wilkie translation from proofs in bounded arithmetic to proofs in propositional logic.

Let M be any countable nonstandard model of true arithmetic and a any nonstandard element of M. We define M_a to be the structure whose domain is the cut in M of numbers subexponential in a, that is

$$M_a = \bigcap_{\varepsilon > 0} \{u \in M : u < 2^{a^\varepsilon}\},$$

and whose language L_a contains a predicate symbol for every bounded subset of M_a which is coded in M.

Theorem 14 *[6] Let Φ_n be a family of CNF formulas whose variables come from a language L.*

Then (Φ_n) has no subexponential size family of $\mathrm{Res}(\log)$ *refutations if and only if every model of the form M_a can be expanded to a model (M_a, L) of $T^2_2(L_a, L)$ in which Φ_a is true.*

Lemma 15 *If $\overline{2\mathrm{VR}^2}$ has subexponential size* $\mathrm{Res}(\log)$ *refutations, then the propositional translation of* $\neg 2{-}\mathrm{Ref}(\mathrm{PK}_1)$ *has subexponential size* $\mathrm{Res}(\log)$ *refutations. Similarly for* $\overline{2\mathrm{VR}^1}$ *and* $1{-}\mathrm{Ref}(\mathrm{PK}_1)$.

Proof We do the $\overline{2\mathrm{VR}^2}$ case.

Suppose that the conclusion of the lemma is false and take any model M_a. Then there is an expansion of M_a to a model of $T^2_2(\Gamma, \Pi, \alpha)$ in which the size a translation of $\neg 2{-}\mathrm{Ref}(\mathrm{PK}_1)$ is true. Hence $2{-}\mathrm{Ref}(\mathrm{PK}_1)(\Gamma, \Pi, \alpha, a)$ is false in the model. Let V, F, G, H, S, T be polynomial time with oracles for Γ, Π, α as given by Theorem 13. Since these are polynomial time, we may add them to the language and our model will still satisfy T^2_2 in this expanded language. But since $2\mathrm{VR}^2(t(a)) \rightarrow 2{-}\mathrm{Ref}(\mathrm{PK}_1)(a)$ is provable in T^2_2, we must have that $2\mathrm{VR}^2(t(a))$ is false in the model. So by the theorem $\overline{2\mathrm{VR}^2}$ does not have subexponential size $\mathrm{Res}(\log)$ refutations. □

Corollary 16 *If $\overline{2\mathrm{VR}^2}$ has subexponential size* $\mathrm{Res}(\log)$ *refutations, then every CNF with a PK_1 refutation of size a*

has a $\mathrm{Res}(\log)$ *refutation subexponential in* a*. Similarly for* $\overline{2\mathrm{VR}^1}$ *and bounded-width CNFs.*

Proof We do the $\overline{2\mathrm{VR}^2}$ case.

Let P be a $\mathrm{Res}(\log)$ refutation of the translation of $\neg 2{-}\mathrm{Ref}(\mathrm{PK}_1)(\Gamma, \Pi, \alpha, a)$ of size subexponential in a, as given by the lemma. Recall that $2{-}\mathrm{Ref}(\mathrm{PK}_1)$ has the form "either there is a witness that Π is not a well-formed refutation of Γ, or there is some clause in Γ which is falsified by α" so the translation of $\neg 2{-}\mathrm{Ref}(\mathrm{PK}_1)$ is the conjunction of translations of "nothing witnesses that Π is not a well-formed refutation of Γ" and "every clause in Γ contains a literal made true by α".

Suppose we have a size a PK_1 refutation Π^* of a formula Γ^*. Suitably coded, we can use these to assign values to all the variables coming from Π and Γ in our $\mathrm{Res}(\log)$ refutation P. This will satisfy all clauses in the first conjunct of $\neg 2{-}\mathrm{Ref}(\mathrm{PK}_1)$ (since Π^* genuinely is a PK_1 refutation of Γ^*), so the first conjunct vanishes. In the second conjunct all the literals will vanish except those from α, and what will be left will be a set of clauses isomorphic to Γ^*, but in these new literals.

Hence our P will have become a refutation of Γ^* of the required size. $\square$

5. Acknowledgment

The authors would like to thank Jan Krajíček, Leszek Kołodziejczyk and Pavel Pudlák for helpful discussions about this material.

References

[1] M. Ajtai. The complexity of the pigeonhole principle. In *29th Annual Symposium on Foundations of Computer Science*, pages 346–355, White Plains, New York, 24–26 Oct. 1988. IEEE.

[2] S. Buss. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[3] S. R. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Annals of Pure and Applied Logic*, 75(1–2):67–77, 12 Sept. 1995.

[4] M. Chiari and J. Krajíček. Lifting independence results in bounded arithmetic. *Archive for Mathematical Logic*, 38(2):123–138, 1999.

[5] J. Krajíček and P. Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschr. f. Mathematikal Logik u. Grundlagen d. Mathematik*, 36(1):29–46, 1990.

[6] J. Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[7] J. Krajíček, A. Skelley, and N. Thapen. NP search problems in low fragments of bounded arithmetic. *The Journal of Symbolic Logic*. To appear.

[8] A. Maciel, T. Pitassi, and A. Woods. A new proof of the weak pigeonhole principle. *Journal of Computer and System Sciences*, 64(4):843–872, 2002.

[9] P. Pudlák. Ramsey's theorem in bounded arithmetic. In E. Borger, H. K. Buning, M. M. Richter, and W. Schonfeld, editors, *Proceedings of Computer Science Logic*, volume 553 of *LNCS*, pages 308–312. Springer-Verlag, 1992.

[10] A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Available at `http://www.mi.ras.ru/~razborov/`, 2003.

[11] N. Segerlind, S. R. Buss, and R. Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM J. Comput*, 33(5):1171–1200, 2004.

[12] D. Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.