The Provably Total Search Problems of Bounded Arithmetic

Alan Skelley* Neil Thapen†

April 29, 2010

Abstract

We give combinatorial principles GI_k, based on k-turn games, which are complete for the class of NP search problems provably total at the kth level T_2^k of the bounded arithmetic hierarchy and hence characterize the $\forall\hat{\Sigma}_1^b$ consequences of T_2^k, generalizing the results of [22]. Our argument uses a translation of first order proofs into large, uniform propositional proofs in a system in which the soundness of the rules can be witnessed by polynomial time reductions between games.

We show that $\forall\hat{\Sigma}_1^b(\alpha)$ conservativity of of $T_2^{i+1}(\alpha)$ over $T_2^i(\alpha)$ already implies $\forall\hat{\Sigma}_1^b(\alpha)$ conservativity of $T_2(\alpha)$ over $T_2^i(\alpha)$. We translate this into propositional form and give a polylogarithmic width CNF $\overline{\mathrm{GI}}_3$ such that if $\overline{\mathrm{GI}}_3$ has small R(log) refutations then so does any polylogarithmic width CNF which has small constant depth refutations. We prove a resolution lower bound for $\overline{\mathrm{GI}}_3$.

We use our characterization to give a sufficient condition for the totality of a relativized NP search problem to be unprovable in $T_2^i(\alpha)$ in terms of a non-logical question about multiparty communication protocols.

Mathematics Subject Classification: 03F30, 68Q15, 03F20

1 Introduction

In the first two sections we give some background, our main result and some applications. The technical proofs follow.

*Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA, alanoman@google.com. Supported in part by Eduard Čech Center grant LC505 and NSERC PDF-313650-2005. This work was partly carried out while the author was at the Institute of Mathematics of the Academy of Sciences of the Czech Republic in Prague and at the Dipartimento di Informatica of the Università degli Studi di Roma "La Sapienza".

†Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, 115 67 Prague 1, Czech Republic, thapen@math.cas.cz. Supported in part by grant AV0Z10190503 and by Eduard Čech Center grant LC505.

1.1 Background

Bounded arithmetic is a name for a collection of fragments of Peano arithmetic in which exponentiation is not a total function and in which the induction axiom is only applied to bounded formulas. We give an overview of the relevant definitions here; for a full introduction see for example [6], [18] or [9]. See [15] for our language and the precise connection between PV and T^0_2. We use the language $\{0, 1, +, \cdot, <, |x|, x\#y, \lfloor \frac{x}{2^y} \rfloor\}$ where $|x| \sim \log_2 x$ is the length of the binary expression for the number x and the function $\#$ is a weak form of exponentiation with $x\#y = 2^{|x|\cdot|y|}$, guaranteeing that the set of lengths in a model is closed under multiplication. We take an algebraic theory BASIC fixing the simple properties of this language.

In a *bounded formula* all quantifiers are bounded by terms in the language. In a *sharply bounded formula* all quantifiers are bounded by terms of the form $|t|$; such quantifiers are called *sharply bounded.* The bounded and sharply bounded quantifers play the roles in bounded arithmetic that unbounded and bounded quantifiers respectively play in classical theories. A $\hat{\Sigma}^b_i$ formula consists of i alternating blocks of bounded quantifiers, with the first one existential, followed by a sharply bounded formula. $\hat{\Pi}^b_i$ formulas are defined dually. We are usually interested in bounded formulas of this strict form, where all the sharply bounded quantifiers come at the end. This strictness is what the $\hat{}$ in $\hat{\Sigma}^b_i$ denotes.

For $i \geq 0$ the theory T^i_2 is BASIC together with the axiom IND(ϕ),

$$[\phi(0) \wedge \forall x{<}a\, (\phi(x) \rightarrow \phi(x+1))] \rightarrow \phi(a),$$

for each $\hat{\Pi}^b_i$ formula ϕ or, equivalently, for each $\hat{\Sigma}^b_i$ formula ϕ. The theory S^i_2 is defined similarly but with induction only up to $|a|$.

This presentation (or rather, a sequent calculus version of it) is what we will use when doing formal proof theory. In practice, however, we will work with an equivalent theory in a richer language. It is possible to define the polynomial time functions on $\mathbb{N}$ as the closure of some basic functions under composition, projection and a kind of polynomial time recursion. The set L_{PV} of PV *function symbols* contains a name for each function defined in this way, and the (first order) theory PV contains the natural defining axioms for these symbols, together with induction for open formulas. T^0_2 is conservative with PV and over this theory, for $i \geq 1$, we may freely use L_{PV} symbols in $\hat{\Sigma}^b_i$ formulas without increasing their expressive power or the power of our induction axioms. We will use PV as our base theory. We will also refer to atomic formulas in the language L_{PV} as PV formulas. When we talk about a polynomial time relation or function in a formal setting, the intended meaning is a PV formula or function symbol.

If α is a new, undefined relation or function symbol, or a tuple of such symbols, then formulas from the *relativized* classes $\hat{\Sigma}^b_i(\alpha)$, $\hat{\Pi}^b_i(\alpha)$ or PV(α) are allowed to use symbols from α freely. We will sometimes write, say, a PV(α) formula in the form $\phi(x, \alpha)$ if we want to emphasize the symbol α or have the possibility of substituting a different symbol or formula for it. The relativized theories $T^i_2(\alpha)$ or PV(α) have axioms for induction or polynomial time recursion for formulas or functions containing the new symbols (as well as axioms bounding the size of any new function symbol by some term in the original language). The new symbols behave analogously to oracles in complexity theory – in particular, the presence of these symbols allows us to prove some separation results for the relativized theories. Our main theorems and their proofs in Sections 3 and 4 are presented in an unrelativized form but relativize without any problems because, from our point of view, all that relativization does is enrich the language and the details of the language are not important for these arguments. We will need the relativized versions for our applications in Section 2.

The main open problem in the area is to show that full bounded arithmetic $T_2 =_{def} \bigcup_i T_2^i$ does not collapse to some finite level T_2^j. This is equivalent to showing that bounded arithmetic does not prove that the polynomial hierarchy collapses [21, 8, 35] (it is known that the relativized bounded arithmetic hierarchy does not collapse [21]). A natural conjecture is that the theories T_2^i are already separated by $\forall\hat{\Pi}_1^b$ formulas, by analogy with the classical theories $\mathrm{I}\Sigma_i$ which are separated by Π_1 consistency statements. However direct consistency arguments will not work, since even strong bounded arithmetic theories are known not to prove the consistency of weak ones [34, 29].

This paper is concerned with what seems to be the most tractable approach to getting more information about the hierarchy, which is to look for a $\forall\hat{\Sigma}_1^b$ separation between the bounded arithmetic theories in the relativized setting. The witnessing theorem method is available to study such sentences: first show that if a $\forall\hat{\Sigma}_1^b(\alpha)$ sentence is provable in a theory, then witnessing it is reducible to finding a witness to some NP property of a combinatorial structure built up out of polynomial time oracle machines; then show a limit to how much information such a witness can give about the oracle [11]. The problem of finding a witness to an NP predicate when one is known to exist is called an NP search problem. TFNP is the class of all such problems. There is a rich variety of subclasses of TFNP, often characterized by the combinatorial lemma which guarantees that solutions to the problems in the class exist. See, for example, [26, 16, 2] for discussions of some of these.

Well-known examples of this connection in bounded arithmetic are S_2^1 and polynomial time functions [6] and T_2^1 and polynomial local search or PLS problems [10]. In [11] the PLS characterization was used to show that $T_2^1(\alpha)$ is not $\forall\hat{\Sigma}_1^b(\alpha)$ conservative over $T_2^2(\alpha)$. This is still the highest $\forall\hat{\Sigma}_1^b(\alpha)$ separation known; in particular it is unknown whether or not $T_2^2(\alpha)$ already proves all the $\forall\hat{\Sigma}_1^b(\alpha)$ consequences of the complete hierarchy. Improving this is closely connected with the problem of finding nice lower bounds for R(log) in propositional proof complexity (see Section 2.2 below).

Various characterizations of search problems corresponding to theories higher in the hierarchy are known, based on reflection principles or on different kinds of Herbrandization [20, 24, 13, 11, 14, 30, 22, 31]. In this paper we generalize and simplify the characterization in [22] and give natural combinatorial principles capturing the $\forall\hat{\Sigma}_1^b$ consequences of the whole hierarchy, which we are able to use in some small applications. We are not able to prove any new separation but we include the resolution lower bound in Section 5 as a partial result in this direction.

This paper was originally circulated in late 2007. Since then two preprints have appeared with related, independent work on characterizing the low complexity consequences of the bounded arithmetic hierarchy, in terms of principles generalizing PLS [4, 5].

Formally we define an *NP search problem* or just *search problem* to be a true $\forall\hat{\Sigma}^b_1$ sentence. Such a sentence, say $\forall x\, \exists y\!<\!t(x)\, \theta(x,y)$ where θ is a PV formula, represents the problem of finding, given x, a *solution* or *witness* $y < t(x)$ such that $\theta(x,y)$ is true.

A search problem $\forall x\, \exists y\!<\!t(x)\, \theta(x,y)$ is *reducible to* a search problem $\forall u\, \exists v < s(u)\, \phi(u,v)$ if there are polynomial time functions f and g such that for all x and v, if $v < s(f(x)) \wedge \phi(f(x),v)$ then $g(x,v) < t(x) \wedge \theta(x, g(x,v))$. If Γ and Δ are classes of search problems then Γ is reducible to Δ, written $\Gamma \leq \Delta$, if every problem in Γ is reducible to a problem in Δ. We write $\Gamma \equiv \Delta$ if both classes are reducible to each other.

In this paper, a *combinatorial principle* will mean a class of search problems arising in a uniform way. It will be given by tuples $\bar{x}$ and $\bar{y}$ of variables called respectively *size parameters* and *witnesses*, a language λ of new relation and function symbols, a PV term t and a PV(λ) formula $\theta(\lambda,\bar{x},\bar{y})$. The search problems in the class are all the sentences of the form $\forall\bar{x}\, \exists\bar{y} < t(\bar{x})\, \theta(\lambda,\bar{x},\bar{y})$ where we substitute in polynomial time machines (or oracle machines, in the relativized setting) for the symbols λ. By an *instance* of a principle we mean a particular assignment to $\bar{x}$ and λ. We will also use it to mean the formula $\exists\bar{y}\!<\!t(\bar{x})\, \theta(\lambda,\bar{x},\bar{y})$ under this assignment. The machines are always given access to the size parameters as extra inputs and "polynomial time" will always mean polynomial in the total length of the size parameters. We will usually describe a combinatorial principle by describing a typical instance.

This is a rather ad hoc way of formalizing search problems. Alternatives would be to place more emphasis on problems being given by oracles (see for example [2]) or by circuits. With our definitions the set $\forall\hat{\Sigma}^b_1(T^k_2)$ of $\forall\hat{\Sigma}^b_1$ sentences provable in T^k_2 is formally a class of search problems. To give an axiomatization of this set of sentences over our base theory PV it is enough to give a class of search problems to which $\forall\hat{\Sigma}{}^b_1(T^k_2)$ is reducible, with each reduction provable in PV.

The propositional proofs we consider will be refutations, or pieces of refutations, using fragments of the propositional calculus PK. A refutation is a sequence of *cedents*, which are sets of formulas that are interpreted as disjunctions; the initial cedents come from the formula we want to refute; each cedent follows from earlier ones by a sound rule; the last cedent is empty and thus unsatisfiable. Sometimes we will also allow the introduction of new cedents as axioms. In Sections 3 and 4 we give formal definitions of the systems we need for our main result.

A *clause* is a cedent which contains only literals, that is, propositional variables or their negations. A *CNF formula* is a set of clauses, which we can think of as a conjunction of disjunctions of literals. We will usually deal with families $\phi_1, \phi_2, \ldots$ or $\pi_1, \pi_2, \ldots$ of formulas or refutations, rather than single ones. The size of a formula or refutation is the total number of symbols it contains. For a formula ϕ_a this will always be quasipolynomial in a, that is, bounded by $2^{|a|^{O(1)}}$. Hence we think of the index a as a size parameter. A *narrow* family of CNFs is one where the size of the clauses is polylogarithmic in the size parameter.

1.2 Main theorem

Our characterization will be in terms of games with two players and a fixed finite number k of turns. The two players A and B take alternate turns, with A going first. Formally a game is given by a k-ary relation G and a size parameter b. The moves are numbers smaller than b and $G(x_1, \ldots, x_k)$ holds if the second player wins in the game with the sequence of moves $x_1, \ldots, x_k$.

We will write *B has a winning strategy for game G* to stand for the sentence

$$\forall x_1 < b\, \exists x_2 < b\, \ldots Q_k x_k < b\, G(x_1, \ldots, x_k)$$

where the last quantifier Q_k is universal if k is odd and existential if k is even. The game can be thought of as being about this sentence, with A trying to find counterexamples to the universal quantifiers and B trying to find witnesses for the existential quantifiers (in the literature on games and quantifiers the second player is often called E to bring out the connection with $\forall$ and $\exists$).

In contrast, by a *polynomial time strategy* for a game we will mean a sequence of polynomial time functions which gives one player's moves explicitly in terms of the other player's past moves.

Definition 1 *Suppose G and H are two k-turn games. We say that G is* polynomial time reducible *to H if there are polynomial time functions $f_1, \ldots, f_k$ such that for all possible sequences of moves $\bar{x}$ in G and $\bar{y}$ in H, if $y_i = f_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1})$ for every odd i and $x_i = f_i(x_1, \ldots, x_{i-1}, y_1, \ldots, y_i)$ for every even i, then $H(\bar{y})$ implies $G(\bar{x})$.*

We can draw a picture of this, here for even k:

$$\begin{array}{lccccc} H: & y_1 & y_2 & y_3 & \cdots & y_k \\ & f_1 \uparrow & f_2 \downarrow & f_3 \uparrow & \cdots & f_k \downarrow \\ G: & x_1 & x_2 & x_3 & \cdots & x_k \end{array}$$

The functions $f_1, \ldots, f_k$ give a reduction if, whenever $\bar{x}$ and $\bar{y}$ are matched as in the picture and Player B wins in H with these moves, then Player B also wins in G.

Notice that this is a natural Herbrandization by polynomial time functions of the sentence "if B has a winning strategy for H, then B also has a winning strategy for G", and in particular that this sentence is implied by the existence of a reduction. Notice also that for $k = 2$, "B has a winning strategy for G" and "B has a winning strategy for H" can be thought of as search problems and our definition of reducibility between games coincides with the definition of reducibility between search problems.

In the following definition a "uniform" sequence means that, for example, the sequence of k-ary relations $G_1, \dots, G_a$ is really a single $(k+1)$-ary relation G with the extra parameter written as an index. Recall also, from the definition of a combinatorial principle, that G and all other polynomial time objects also have access to the size parameters as unwritten inputs.

We will often treat strategies and reductions, which are formally sequences of functions, as coded by a single function. The details of the coding will not matter.

Definition 2 *An instance of the* k-game induction principle GI_k *is given by size parameters a and b, a uniform sequence $G_1, \dots, G_a$ of polynomial time relations, polynomial time functions U and V and a uniform sequence $W_1, \dots, W_{a-1}$ of polynomial time functions.*

The instance $\mathrm{GI}_k(G, U, V, W, a, b)$ states that, interpreting each G_i as a k-turn game in which the moves are bounded by b, the following things cannot all be true:

1. *U is a winning strategy for B in G_1;*

2. *V is a winning strategy for A in G_a;*

3. *For each i, W_i gives a reduction of G_{i+1} to G_i.*

Often we will not distinguish the parameters a and b and will use a single size parameter for the number of games and the bound on the moves.

The principle is $\forall\hat{\Sigma}^b_1$. It is provable in T^k_2 by induction up to a on i in the formula "Player B has a winning strategy in game G_i" which is $\hat{\Pi}^b_k$ (in fact, 1, 2 and 3 are a natural Herbrandization by polynomial time functions of the negation of this instance of the induction scheme). Hence for fixed polynomial time G, U, V, W we have $\mathrm{GI}_k(G, U, V, W) \in \forall\hat{\Sigma}^b_1(T^k_2)$, immediately giving one direction of our main result:

Theorem 3 *For all $k \geq 1$,* $\mathrm{GI}_k \equiv \forall\hat{\Sigma}^b_1(T^k_2)$ *provably in* PV.

"Provably" here means, for example, that for each search problem σ in the class on the left there is a search problem τ in the class on the right such that the sentence expressing the reduction of σ to τ is provable in PV.

The bulk of the paper is a proof of the other direction. It goes via a reflection principle $1\mathrm{-Ref}(\mathrm{PK}^0_k)$ for a certain propositional proof system. The idea is that the first order T^{k+2}_2 proof that a search problem is total is used as a recipe for building a "program" that witnesses the problem. The program is in fact a complicated sequence of game reductions, which we think of as something like the diagram in Section 1.3 below. We use our propositional proof system as a convenient notation for describing this sequence.

In Sections 3 and 4 we show respectively:

Theorem 4 *For all $k \geq 0$,* $1\mathrm{-Ref}(\mathrm{PK}^0_k) \leq \mathrm{GI}_{k+2}$ *provably in* PV.

Theorem 5 *For all $k \geq 0$,* $\forall\hat{\Sigma}^b_1(T^{k+2}_2) \leq 1\mathrm{-Ref}(\mathrm{PK}^0_k)$ *provably in* PV.

This leaves the case of T^1_2. We can do this directly using the existing characterization of $\forall\hat{\Sigma}^b_1(T^1_2)$ in terms of PLS problems:

Theorem 6 $\forall\hat{\Sigma}^b_1(T^1_2) \leq \mathrm{GI}_1$ *provably in* PV.

Proof GI_1 is more naturally thought of as a kind of iteration principle than as a statement about one turn games. An instance is given by a sequence of predicates H_i (standing for "A wins game $a+1-i$"), a sequence of functions f_i (standing for the function comprising the reduction W_{a-i}) and a single number e (standing for A's winning strategy in game a). The principle states that if $H_1(e)$ and $H_i(x) \to H_{i+1}(f_i(x))$ for all i and x, then $H_a(x)$ for some x (all suitably bounded).

Consider an instance (N, C, a) of PLS with domain $[0, a)$, neighbourhood function N and cost function C, where the costs lie in $[1, a)$. This is reducible to our iteration principle by defining $H_i(x)$ to be $C(x) \leq a-i$, $f_i(x)$ to be $N(x)$ and e to be 0. □

1.3 Discussion

We can draw a suggestive picture of our principle, here for the case $k = 4$.

$$\begin{tikzcd}
G_1 : & & x_1^1 \arrow[r] & x_2^1 \arrow[d] & x_3^1 \arrow[r] & x_4^1 \arrow[d] \\
G_2 : & & x_1^2 \arrow[u] & x_2^2 \arrow[d] & x_3^2 \arrow[u] & x_4^2 \arrow[d] \\
\vdots & & \vdots \arrow[u] & \vdots \arrow[d] & \vdots \arrow[u] & \vdots \arrow[d] \\
G_a : & {} \arrow[r] & x_1^a \arrow[u] & x_2^a \arrow[r] & x_3^a \arrow[u] & x_4^a
\end{tikzcd}$$

Each arrow represents a polynomial time function. The horizontal arrows at the top represent B's strategy in G_1, the horizontal arrows at the bottom represent A's strategy in G_a and each row of vertical arrows represents a game reduction. It is tempting to try to understand our search problems in terms of iterating a polynomial time function along the arrows in the diagram, in the spirit of the characterization of the provably recursive functions of fragments of Peano arithmetic in terms of iterating a primitive recursive function using an ordinal below ε_0. In this way PV or S^1_2 would correspond to iterations of length polylogarithmic in the parameters; T^1_2 would somehow correspond to iterations of length quasipolynomial in the parameters; the extra strength of T^2_2 over T^1_2 would come not from the iteration being longer, but from being allowed to revisit, in a controlled way as you follow the arrows back down the structure, the values calculated on the first upwards pass (it is not possible simply to remember all the past values, because of the bound on the size of the moves); and so on. However, actually iterating a polynomial time function a times, represented by just the arrows going up the first column, is already equivalent to computing something in PSPACE, which is certainly stronger than PLS (at least in the relativized case). Nevertheless the connection seems to be worth exploring. See for example the characterization in Section 2.3 below, or [3] where low bounded arithmetic theories are characterized in terms of the lengths of the intervals on which they can prove a well-foundedness principle.

In [22] it was shown that $\forall\hat{\Sigma}^b_1(T^2_2)$ can be characterized by the principles CPLS, VR-totality and 2VR(log)-totality and $\forall\hat{\Sigma}^b_1(T^3_2)$ by 2VR-totality. Our results here generalize these to higher levels in the hierarchy; however there are some things from there which we do not explore further. One is that instances of CPLS seem to correspond to instances of GI_2 with the extra property that the second function f_2 in all the game reductions is the identity. We do not know whether any generalization of this can hold for GI_k. Also, while VR-totality and 2VR-totality are similar to GI_2 and GI_3, a natural definition of 2VR(log)-totality in our new setting would be: like GI_3, except that now there are only $|a|$ many games and that where in GI_3 we produce moves for the game G_{i+1} by querying moves in G_i, now we are allowed to query moves in *two* instances of G_i. We do not know how to generalize this characterization. One problem is that, for games with several turns, it seems to be important in which order we query the two instances at each turn; but it is not clear what a good order should be. This situation appears to be connected to the "schedules" of [31].

2 Applications

2.1 A "no gap" theorem

We can apply our characterization to give a strengthening of a theorem of Chiari and Krajíček [12].

Theorem 7 *For $i \geq 1$ and $k \geq 1$, if $T^{i+1}_2(\alpha)$ is $\forall\hat{\Sigma}^b_k(\alpha)$ conservative over $T^i_2(\alpha)$ then full bounded arithmetic $T_2(\alpha)$ is $\forall\hat{\Sigma}^b_k(\alpha)$ conservative over $T^i_2(\alpha)$.*

Proof The case for $k \geq 2$ was proved in [12] so we just need to deal with the case $k = 1$. It is enough to show that, for all i, conservativity of $T^{i+1}_2(\alpha)$ over $T^i_2(\alpha)$ implies conservativity of $T^{i+2}_2(\alpha)$ over $T^{i+1}_2(\alpha)$, since the result then follows by induction. So suppose that $T^{i+1}_2(\alpha)$ is $\forall\hat{\Sigma}^b_1(\alpha)$ conservative over $T^i_2(\alpha)$. We will show that GI_{i+2} is provable in $T^{i+1}_2(\alpha)$ for instances involving α. We will do the case for even i. The argument for odd i is dual to this.

Let (G, U, V, W, a, b) be an instance of GI_{i+2}, which may involve our undefined relation α. We will use this to define an "instance" (G', U', V', W', a, b) of GI_{i+1}: for each j, $G'_j(x_1, \ldots, x_{i+1})$ is $\exists x_{i+2} < b\, G_j(x_1, \ldots, x_{i+2})$ and the functions U', V', W' are given by the natural restrictions of strategies or reductions for $i + 2$ turn games to the first $i + 1$ turns.

Treating G', U', V' and W' as undefined symbols, by our conservativity assumption $\mathrm{GI}_{i+1}(G', U', V', W')$ is provable in $T^i_2(G', U', V', W')$. Substituting the definitions of (G', U', V', W') into this proof will increase the quantifier complexity by at most one (here we are relying on the fact that sharply bounded collection is provable in $T^1_2(\alpha)$, so non-strict $\Sigma^b_1(\alpha)$ formulas are equivalent to $\hat{\Sigma}^b_1(\alpha)$ formulas). Hence $\mathrm{GI}_{i+1}(G', U', V', W')$ is provable in $T^{i+1}_2(\alpha)$.

To complete the proof it is enough to show (over our base theory PV(α)) that if there is a witness to GI_{i+1} for (G', U', V', W', a, b) then there is a witness to GI_{i+2} for (G, U, V, W, a, b).

Suppose that $(x_1, \ldots, x_{i+1})$ is a sequence of moves in G'_1 in which B plays according to the strategy U' but loses, so that $\neg G'_1(x_1, \ldots, x_{i+1})$. U' is a restriction of U, so we can extend the sequence to $(x_1, \ldots, x_{i+2})$ in which B plays according to U. But by the definition of G'_1, B must lose G_1 with these moves. Hence $(x_1, \ldots, x_{i+2})$ witnesses our instance of GI_{i+2}.

Suppose that $(x_1, \ldots, x_{i+1})$ is a sequence of moves in G'_a in which A plays according to the strategy V' but loses, so that $G'_a(x_1, \ldots, x_{i+1})$. Then by the definition of G'_a we can extend the sequence by one move x_{i+2} for B such that $G_a(x_1, \ldots, x_{i+2})$. But in this sequence A is playing according to V, but nevertheless loses G_a. Hence $(x_1, \ldots, x_{i+2})$ witnesses our instance of GI_{i+2}.

Suppose that $(x_1, \ldots, x_{i+1})$ and $(y_1, \ldots, y_{i+1})$ are sequences of moves in G'_j and G'_{j+1} which are matched by the reduction W'_j, with $G'_j(x_1, \ldots, x_{i+1})$ but $\neg G'_{j+1}(y_1, \ldots, y_{i+1})$. This looks like

$$\begin{array}{rcccc} G'_j: & x_1 & x_2 & \ldots & x_{i+1} \\ & f_1 \uparrow & f_2 \downarrow & \ldots & f_{i+1} \uparrow \\ G'_{j+1}: & y_1 & y_2 & \ldots & y_{i+1} \end{array}$$

where $f_1, \ldots, f_{i+1}$ comprise the reduction W'_j, which is a restriction of the reduction W_j. Say W_j extends W'_j with one more function f_{i+2}. By the definition of G'_j there is x_{i+2} such that $G_j(x_1, \ldots, x_{i+2})$. But if we let $y_{i+2} = f_{i+2}(x_1, \ldots, x_{i+2}, y_1, \ldots, y_{i+1})$ then by the definition of G'_{j+1} we must have $\neg G_{j+1}(y_1, \ldots, y_{i+2})$. But the xs and ys are matched by W_j, so again we have a witness to our instance of GI_{i+2}. □

2.2 A candidate principle hard for R(log)

The *depth* of a propositional refutation is the maximum number of alternations of connectives in any formula in it. We can also say that a refutation has depth $k + \frac{1}{2}$ if there are k alternations of connectives with arbitrary fan-in and then an innermost level of connectives with "small" fan-in. Here "small" means logarithmic in the size of the refutation; we are usually dealing with quasipolynomial size refutations, so small will mean polylogarithmic in our size parameter. PK_k is PK restricted to depth k. Resolution is then PK_0 and R(log) is $\mathrm{PK}_{\frac{1}{2}}$, except that we will also allow axioms of the form $\{p, \neg p\}$ to appear in R(log) refutations, since these seem to be necessary for our translation in this section. R(log) was introduced in [19] as a strengthening of resolution to a system corresponding to the theory $T^2_2(\alpha)$ via the Paris-Wilkie translation of first order proofs into propositional proofs. In general, depth $k + \frac{1}{2}$ corresponds to $T^{k+2}_2(\alpha)$ [17, 19]; we come back to this in Section 4.

A current problem in proof complexity is to prove lower bounds on the size of R(log) refutations of some family of CNFs which have small refutations of some constant depth (it is known for example that there is no small R(log) proof of the pigeonhole principle, but this is because it has no small proofs of any constant depth [23, 28]). A super-quasipolynomial lower bound for a narrow family of such CNFs would imply a $\forall\hat{\Sigma}^b_1(\alpha)$ separation of $T_2(\alpha)$ from $T^2_2(\alpha)$. Recent work has looked at lower bounds for the weak pigeonhole principle from $2n$ to n. It was shown in [33] 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 [32] to $\varepsilon \log n/\log\log n$; but it is known to be refutable with quasipolynomial size in R(log) [25].

The narrow CNF $\overline{\mathrm{GI}}_3(a)$ is a propositional translation of the negation of a size a instance of GI_3. We write it out explicitly at the start of Section 5 below. It has polynomial size PK_1 refutations (see Lemma 22); these are essentially translations of the proof of GI_3 using $\hat{\Pi}^b_3$ induction. However, by the next theorem it is unlikely that $\overline{\mathrm{GI}}_3$ has small R(log) refutations. We are able to prove an exponential lower bound on the size of resolution refutations of $\overline{\mathrm{GI}}_3$, which we include as Section 5 of the paper.

We should remark that the role GI_3 is playing in the current section is really that of a reflection principle for PK_1. We believe that an advantage of our principle is that it has a more transparent combinatorial meaning than straight PK_1 reflection, which we exploit in our lower bound argument and which may be useful for proving lower bounds in stronger systems. However resolution lower bounds for reflection for resolution itself are already known [1] (using a different method); resolution is a subsystem of PK_1, so this already implies a lower bound for PK_1 reflection.

Theorem 8 *If $\overline{\mathrm{GI}}_3$ has quasipolynomial size* R(log) *refutations, then so does any narrow CNF with quasipolynomial size bounded depth refutations.*

The proof is essentially by Krajíček's correspondence between propositional lower bounds and expansions of non-standard models [19] together with a version of Theorem 7. However we give the outline of a proof-theoretic argument, since this paper already contains most of the necessary machinery.

Lemma 9 *Let $\Phi(\pi)$ be $\forall\hat{\Pi}^b_1(\pi)$ and let $i \geq 1$. If $T^{i+1}_2(\alpha)$ is $\forall\hat{\Sigma}^b_1(\alpha)$ conservative over $T^i_2(\alpha,\pi)+\Phi(\pi)$ then $T^j_2(\alpha)$ is $\forall\hat{\Sigma}^b_1(\alpha)$ conservative over $T^i_2(\alpha,\pi)+\Phi(\pi)$ for all $j > i$.*

Proof We use induction on j. Suppose that $T^i_2(\alpha,\pi)+\Phi(\pi)$ proves all $\forall\hat{\Sigma}^b_1(\alpha)$ consequences of $T^j_2(\alpha)$.

We will use the same trick as in Theorem 7. We know that $T^i_2(\beta,\pi)+\Phi(\pi)$ proves the totality of every GI_j search problem with an oracle β. By taking these proofs and substituting a $\hat{\Sigma}^b_1(\alpha)$ formula for β, as in Theorem 7 we can show that $T^{i+1}_2(\alpha,\pi)+\Phi(\pi)$ proves the totality of every GI_{j+1} search problem with an oracle α. Now let $\sigma(\alpha)$ be any $\forall\hat{\Sigma}^b_1(\alpha)$ consequence of $T^{j+1}_2(\alpha)$. By our main theorem, we have

$$T^{i+1}_2(\alpha,\pi) \vdash \neg\Phi(\pi) \vee \sigma(\alpha).$$

By Parikh's theorem we may treat the right hand side as a $\forall\hat{\Sigma}^b_1(\alpha,\pi)$ sentence. We can also code two oracles by one, so may think of α and π as a single oracle.

Hence we can apply our starting assumption to get

$$T_2^i(\alpha, \pi, \pi') + \Phi(\pi') \vdash \neg\Phi(\pi) \vee \sigma(\alpha)$$

where π' is a new oracle symbol. Identifying π and π' in this proof gives the result. $\square$

Lemma 10 *Let $\forall b\, \phi(\pi_b)$ express that the oracle π gives a sequence of quasipolynomial size* R(log) *refutations π_b of $\overline{\mathrm{GI}}_3(b)$, where ϕ is $\hat{\Pi}_1^b(\pi)$ (for how we can think of the structure of a family of proofs or formulas as being given by oracles, see Sections 3 and 4). Then for any problem Γ in* GI_3 *in which the polynomial time functions and relations are allowed to use oracles α,*

$$T_2^2(\alpha, \pi) \vdash \forall a\, [\phi(\pi_a) \to \mathrm{GI}_3(\Gamma, a)].$$

Hence $T_2^2(\alpha, \pi) + \forall b\, \phi(\pi_b)$ proves all $\forall\hat{\Sigma}_1^b(\alpha)$ consequences of $T_2^3(\alpha)$ and thus, by the previous lemma, also of $T_2^j(\alpha)$ for any j.

Proof Fix a. Our instance (Γ, a) of GI_3 defines an assignment β to the variables of $\overline{\mathrm{GI}}_3(a)$. This assignment is polynomial time with oracle α.

Suppose that $\phi(\pi_a)$ is true and $\mathrm{GI}_3(\Gamma, a)$ is false, so that every clause in $\overline{\mathrm{GI}}_3(a)$ is satisfied by β. Then we can prove by induction on i that for every i lines $1, \ldots, i$ of π_a are satisfied by β, since this inductive hypothesis is $\hat{\Pi}_2^b(\beta, \pi)$ and the inductive step follows from the soundness of the rules of R(log). But this is a contradiction when we get to the empty clause at the end of π_a. $\square$

Proof of Theorem 8 Let $k \in \mathbb{N}$. Let $\forall a \phi(\pi_a)$ express that the oracle π gives a sequence of quasipolynomial size R(log) refutations of $\overline{\mathrm{GI}}_3(\mathrm{a})$, with $\phi \in \hat{\Pi}_1^b(\pi)$. Let $\forall b \psi(\rho_b, \tau_b)$ express that the oracle ρ gives a sequence of quasipolynomial size PK_k refutations ρ_b of a sequence of narrow CNFs τ_b given by an oracle τ, with $\psi \in \hat{\Pi}_1^b(\rho, \tau)$. Let α be an assignment to all the variables of $\tau_1, \tau_2, \ldots$. By a standard reflection argument, as in Lemma 10,

$$T_2^{k+2}(\rho, \tau, \alpha) \vdash \forall b\, [\psi(\rho_b, \tau_b) \to (\tau_b \text{ has a false clause under } \alpha)].$$

Because the formulas τ_b are narrow, the formula "τ_b has a false clause under α" is $\hat{\Sigma}_1^b(\tau, \alpha)$ and so by Lemma 10,

$$T_2^2(\rho, \tau, \alpha, \pi) + \forall a\, \phi(\pi_a) \vdash \forall b\, [\neg\psi(\rho_b, \tau_b) \vee (\tau_b \text{ has a false clause under } \alpha)].$$

Rewriting and applying Parikh's theorem, for some term $t(b)$,

$$T_2^2(\rho, \tau, \alpha, \pi) \vdash \forall b\, [\neg\forall a < t(b)\, \phi(\pi_a) \vee \neg\psi(\rho_b, \tau_b) \vee (\tau_b \text{ has a false clause under } \alpha)].$$

Now we translate this $T_2^2(\rho, \phi, \alpha, \pi)$ proof into a R(log) refutation whose variables correspond to the bits of our oracles. This can be done by a version of Theorem 19 below, except that rather than using sharply bounded sentences directly as propositional variables, we translate such sentences into statements about $|b|^{O(1)}$ depth decision trees in the bits of the oracles. The statement that such a tree accepts or rejects can be written as a clause of small conjunctions. Rather than introducing auxiliary clauses, we derive the required relationships between translations of sharply bounded sentences using the rules and axioms of R(log). Hence, for each b, there is a quasipolynomial size (in b) R(log) refutation Π of the translation

$$\langle \forall a < t(b)\, \phi(\pi_a) \rangle + \langle \psi(\rho_b, \tau_b) \rangle + \langle \text{ all clauses of } \tau_b \text{ are true under } \alpha \ \rangle.$$

Now suppose we actually have, in the standard model, quasipolynomial size R(log) refutations π_a of $\overline{\mathrm{GI}}_3(a)$ and quasipolynomial size PK_k refutations ρ_b of our formulas τ_b (these refutations are not required to be uniform). We use these to assign 0/1 values to the variables in Π arising from π, ρ and τ. This will satisfy the first two sets of clauses above and will simplify the last set of clauses into a CNF, in variables from α, which is isomorphic to τ_b. This gives the required R(log) refutation of τ_b. □

2.3 A non-logical characterization

We can use our characterization to give a sufficient condition for the unprovability of a search problem in the relativized version of T^k_2 in terms of a problem in multiparty communication complexity. We define the problem using polynomial time machines, but it would be as interesting to consider machines of unlimited power but which can only send polynomially many bits as messages and can only query, adaptively, polynomially many bits of their oracle.

We are interested in the following situation. Fix a size parameter a and a number of rounds k. Polynomial time will mean polynomial in $|a|$. We are given an exponential length (in $|a|$) sequence of polynomial time machines $M_1, \ldots, M_b$. Each machine M_i has access to its own oracle α_i. The machines trust each other, but we can think of all oracles $\alpha_1, \ldots, \alpha_b$ as being controlled by a single adversary. Each machine M_i can communicate only with its neighbours M_{i-1} and M_{i+1}. Each message can contain polynomially many bits, but messages can only be passed according to strict rules. First M_1 sends one message down to M_2, who then processes it and sends a message down to M_3, and so on until M_{b-1} sends a message to M_b. This is the first round. In the second round M_b sends a message up to M_{b-1}, who processes it and then sends a message up to M_{b-2}, and so on. There are k rounds of communication in total; each odd numbered round consists of a series of messages being sent down from one machine to the next, in order; each even numbered round is similar, but going up. The polynomial bound on the running time of the machines only applies to time spent processing messages, not to time spent waiting for a message.

We say that a communication protocol for this system witnesses an instance $\exists v < a\, \phi(a, v, \alpha)$ of a relativized search problem if some machine M_i eventually shows either that its own oracle satisfies the search problem or that its oracle is different from one of its neighbours' oracles. That is, if some M_i either outputs a witness v such that $\phi(a, v, \alpha_i)$ or outputs a place where α_i is different from α_{i+1} or α_{i-1}.

For example, there is a protocol that witnesses the pigeonhole principle "α is not an injection from $a+1$ into a" in polynomially many (in $|a|$) rounds (the idea for the protocol comes from the development of counting in Frege systems and the related theory U_1^1 [7, 18]). For simplicity suppose that a is a power of 2. There are $a+1$ machines, one for each pigeon, and the key observation is that machine M_i is always able to answer the question "what is $|\{j \le i : \alpha_j(j) \in [c,d]\}|$?" for any interval $[c,d]$ by asking the same question, with $i-1$ in the place of i, to machine M_{i-1} and then adding one to the answer or not depending on the value of $\alpha_i(i)$. So in two rounds M_{a+1} can ask "what is $|\{j \le a+1 : \alpha_j(j) \in [1,a]\}|$?" and necessarily get the answer $a+1$. It then asks about the intervals $[1,a/2]$ and $[a/2+1,a]$ and the two answers must add up to $a+1$, so one of the intervals must have more than $a/2$ pigeons mapping into it. M_{a+1} chooses that interval and then continues by binary search. Eventually this locates a hole c such that for at least two is we have $\alpha_i(i)=c$. There is now one final round of communication. This first passes along the question "Does your pigeon map to c?" until it finds one such i. Then it continues, passing along the question "I think pigeon i maps to c. Does your pigeon also map to c?" This must eventually find either a collision between pigeons in one oracle, or two neighbouring oracles which disagree about pigeon i.

Theorem 11 *Suppose that $T_2^k(\alpha)$ proves that $\forall u\, \exists v < t(u)\, \phi(u,v,\alpha)$ is a total search problem. Then the problem can be witnessed by a protocol as described above, in k rounds.*

Proof Suppose we want to witness $\exists v < t(a)\, \phi(a,v,\alpha)$. By our characterization this problem is reducible to an instance of GI_k in which all polynomial time machines have access to an oracle for α. Say the instance consists of games $G_1,\dots,G_b$, strategies U and V and reductions W. We will do the case where k is even, the case for odd k is similar. Our protocol will differ from the description above in that the first message is passed "up" rather than "down", but this is unimportant.

We will describe the protocol for a machine M_i with $1<i<b$. It will make use of the reduction of G_i to G_{i-1}:

$$\begin{array}{rcccc} G_{i-1}: & x_1 & x_2 & \dots & x_k \\ & f_1\uparrow & f_2\downarrow & \dots & f_k\downarrow \\ G_i: & y_1 & y_2 & \dots & y_k \end{array}$$

All relations and functions are given by polynomial time oracle machines. We will write the oracle being used as a superscript. The protocols for M_1 and M_b will be similar, but rather than sending messages respectively up or down they will use the strategies U and V to obtain replies to these messages.

In the first round M_i gets a message y_1 from M_{i+1}. It calculates $x_1 := f_1^{\alpha_i}(y_1)$, using its own oracle α_i to answer queries made by f_1. It then sends x_1 to M_{i-1}. In the second round M_i gets a message x_2 from M_{i-1}. It calculates $y_2 := f_2^{\alpha_i}(x_1,x_2,y_1)$, again using α_i, and sends y_2 to M_{i+1}.

The protocol carries on like this for all rounds up to and including round $k-1$. Then in the last round M_i gets a pair (x_k,r) of messages from M_{i-1}, where r is a record of the replies that α_{i-1} gave when M_{i-1} ran a computation of $G_{i-1}^{\alpha_{i-1}}(x_1, \ldots, x_k)$. M_i then itself runs a computation of $G_{i-1}^{\alpha_i}(x_1, \ldots, x_k)$, using the oracle α_i. The answer must be the same as the one M_{i-1} got; otherwise M_i has found a place where α_i is different from α_{i-1} and can halt and output this.

M_i calculates $y_k := f_k^{\alpha_i}(x_1, \ldots, x_k, y_1, \ldots, y_{k-1})$. M_i then runs a computation of $G_i^{\alpha_i}(y_1, \ldots, y_k)$, recording the replies made by α_i as a string s.

Now if $G_{i-1}^{\alpha_i}(x_1, \ldots, x_k)$ is true and $G_i^{\alpha_i}(y_1, \ldots, y_k)$ is false then M_i has a solution to an instance $(G, U, V, W, b)^{\alpha_i}$ of GI_k. So by our assumption M_i is able to compute from this some $v < t(a)$ such that $\phi(a, v, \alpha_i)$ and can halt and output this. Otherwise, M_i sends (y_k, s) to M_{i+1}.

If no machines ever finds a difference between neighbouring oracles, then by induction down the sequence of machines there must be some i for which $G_{i-1}^{\alpha_i}(x_1, \ldots, x_k)$ is true and $G_i^{\alpha_i}(y_1, \ldots, y_k)$ is false. So the protocol always succeeds. □

3 Witnessing soundness with game reductions

We define a propositional proof system PK^0. We will show that the soundness of the inferences in this system can be witnessed by a polynomial time reduction between games representing the lines in a proof. For this reason, the important thing about the way the system is defined is that some of the rules involve manipulating formulas by adding or removing only one literal at a time, as we will be able to check the truth of a literal as part of a polynomial time strategy, where we could not check an arbitrary formula. The superscript 0 is meant to indicate the importance of literals, that is, of level 0 formulas.

Definition 12 PK^0 *is a Tait-style calculus (see e.g. [9]). That is, a* PK^0 *proof consists of a sequence of cedents and the cedents are formally sets of formulas rather than sequences or multisets.*

Every formula has a level. *Level* 0 *formulas are propositional variables or their negations. A level* $i+1$ *disjunction is a pair consisting of a label "disjunction" and a set of level* i *conjunctions. Conjunctions are similar. Because of this strict way in which the levels formally operate, we will write* (ϕ) *as shorthand to indicate a formula that has been padded out by singleton disjunctions and conjunctions to raise it to the level we need.*

We list the rules of the system. Below, p *always stands for a single literal,* F *for a single formula and* X*,* Y*,* Γ *and* Δ *for sets of formulas. The principal or auxiliary formulas of a rule are allowed to occur also in the set* Γ *of side formulas.*

1. **Padding introduction**

$$\frac{\Gamma, p}{\Gamma, (p)}$$

That is, we can raise the complexity of p *to any level* i *we choose, by padding out with singleton conjunctions and disjunctions. We can also choose whether to make it into a level* i *disjunction or a level* i *conjunction.*

2. **Padding elimination**

$$\frac{\Gamma, (p)}{\Gamma, p}$$

3. **Internal $\bigwedge$-introduction**

$$\frac{\Gamma, \Phi[\bigwedge X] \qquad\qquad \Gamma, p}{\Gamma, \Phi[\bigwedge X \wedge (p)]}$$

The square bracket notation here indicates that, thinking of formulas as trees, we take a particular subtree $\bigwedge X$ of the tree representing the formula Φ and replace this single subtree with the tree for $\bigwedge Y$ where Y is the set $X \cup \{(p)\}$, where (p) represents the literal p padded out to the same level as the disjunctions in X (recall that for Φ to be well-formed, X must be a set of disjunctions, all of the same level).

4. **Internal $\bigvee$-elimination**

$$\frac{\Gamma, \Phi[\bigvee X \vee (p)]}{\Gamma, \Phi[\bigvee X], p}$$

5. **Internal weakening**

$$\frac{\Gamma, \Phi[\bigvee X]}{\Gamma, \Phi[\bigvee Y]}$$

where $X \subseteq Y$

6. **Internal $\bigwedge$-elimination**

$$\frac{\Gamma, \Phi[\bigwedge X]}{\Gamma, \Phi[\bigwedge Y]}$$

where $X \supseteq Y$

7. **Weakening**

$$\frac{\Gamma}{\Delta}$$

where $\Gamma \subseteq \Delta$

8. **$\bigwedge$-introduction**

$$\frac{\Gamma, F}{\Gamma, \bigwedge\{F\}}$$

That is, a level i disjunction F can be padded to a singleton level $i + 1$ conjunction.

9. **$\bigwedge$-elimination**

$$\frac{\Gamma, \bigwedge\{F\}}{\Gamma, F}$$

10. **Resolution**

$$\frac{\Gamma, p \qquad\qquad \Gamma, \neg p}{\Gamma}$$

Now fix $k \in \mathbb{N}$. We define PK^0_k to be PK^0 with all formulas limited to literals, conjunctions of level k or below and disjunctions of level $k-1$ or below. So PK^0_0 is just resolution (if we think of cedents as clauses) and PK^0_1 is similar to the system of resolution with unbounded conjunctions.

Definition 13 *A k-*formula table *is given by a relation Q and parameters $a_0, \ldots, a_k$. We use the interval $[0, 2a_0)$ as a set of names for a_0 many propositional variables and their negations. We use the interval $[2a_0, 2a_0 + a_1)$ as a set of names for a_1 many level 1 conjunctions. The relation Q, given names for a level 1 conjunction and a literal, tells you whether the literal belongs to the conjunction or not. Similarly with the interval $[2a_0 + a_1, 2a_0 + 2a_1)$ and the level 1 disjunctions, and so on for higher level conjunctions and disjunctions.*

Definition 14 *A* polynomial time PK^0_k derivation *is given by a tuple (Q, R, S, T, f) of polynomial time relations and functions and a size parameter a. Strictly speaking, (Q, R, S, T, f) by themselves define a polynomial time uniform family of PK^0_k derivations; we supply a parameter a and then the relations and functions, taking the parameter as an extra, unwritten input, define the structure of a derivation (of size exponential in $|a|$, or equivalently quasipolynomial in a) as follows:*

Q gives a k-formula table, giving the structure of all the formulas which appear in the derivation. Q also comes with polynomial time functions giving the parameters $a_0, \ldots, a_k$ of the table.

R gives a binary relation, which can be thought of as a table with rows $R_1, \ldots, R_r$ where each R_i is a set of formulas representing an initial cedent of the derivation. R comes with a polynomial time function giving the parameter r. Similarly for S and the internal cedents and T and the final cedents of the derivation.

f gives the structure of the proof. For each internal cedent, it gives the cedent or cedents from which it was derived. It gives the rule used in the derivation, and the principal and auxiliary formulas (except in the case of the normal weakening rule, which has no auxiliary formulas). If a rule refers to a particular subformula of a formula, it gives a path down through the formula to that subformula (thinking of a formula as a tree). Note that a literal that has been padded out is a special case of this: if it is part of the definition of a rule that a subformula (p) is a padded literal, then f should describe the path from (p) down to p.

The formula expressing that the derivation (Q, R, S, T, f, a) is well-formed is $\hat{\Pi}^b_1(Q, R, S, T, f, a)$.

We will give a very simple example, a derivation that contains a single internal $\bigwedge$-introduction inference.

R_1	$\Gamma, \Phi[\bigwedge X]$
R_2	Γ, p
S_1	$\Gamma, \Phi[\bigwedge X]$
S_2	Γ, p
S_3	$\Gamma, \Phi[\bigwedge X \wedge (p)]$
T_1	$\Gamma, \Phi[\bigwedge X \wedge (p)]$

Here $f(\text{“}S_1\text{”})$ and $f(\text{“}S_2\text{”})$ will say that these cedents are copies of the initial cedents R_1 and R_2. $f(\text{“}S_3\text{”})$ will say S_3 is derived by $\bigwedge$-introduction from S_1 and S_2; it will say that the principal formula is $\Phi[\bigwedge X \wedge (p)]$ and the auxiliary formulas are $\Phi[\bigwedge X]$ in S_1 and p in S_2; it will give, as a list of subformulas, the path from the root of the formula-tree of $\Phi[\bigwedge X]$ to the subformula $\bigwedge X$; and it will give, as lists of subformulas, the path from the root of the formula-tree of $\Phi[\bigwedge X \wedge (p)]$ to its subformula $\bigwedge X \wedge (p)$ and the path from $\bigwedge X \wedge (p)$ to its subformula p. $f(\text{“}T_1\text{”})$ will say that T_1 is a copy of the internal cedent S_3.

Recall that a *narrow CNF* is a set of clauses (i.e. of sets of single literals) where each clause only contains polynomially many literals (in $|a|$). The statement that such a CNF is false is $\hat{\Sigma}^b_1$, if the clauses can be listed by a polynomial time function. The index 1 here is the reason for the 1 in $1\text{−Ref}(\text{PK}^0_k)$ below. We could define $2\text{−Ref}(\text{PK}^0_k)$ as a similar reflection principle for CNFs of arbitrary width.

Definition 15 $1\text{−Ref}(\text{PK}^0_k)$ *is a search problem in the language* $(F, \alpha, Q, R, S, T, f)$ *with a size parameter* a. *It expresses that there is always a witness that one of the following things is false:*

1. *F is a narrow CNF, given by a polynomial time function which, given the number of a clause, lists the contents of that clause;*
2. *(Q, R, S, T, f, a) is a well-formed PK^0_k refutation – that is, a derivation where the set T of final clauses consists of a single, empty cedent;*
3. *The number r of clauses in F is the same as the number of cedents in R, and for each $i \leq r$ the ith clause of F is a subset of the ith clause of R;*
4. *Every clause in F is satisfied by the assignment α.*

Note that this defines a $\forall\hat{\Sigma}^b_1$ principle.

We want to reduce $1\text{−Ref}(\text{PK}^0_k)$ to GI_{k+2}, so will need to describe several reductions between games to give our instances of GI_{k+2}. It is convenient to think of the functions f_i that form a reduction of a game G to a game H as components of an explicit winning strategy for the second player D in a certain $2k$-turn, two player *reduction game* played between D and an adversary C. We can draw the *schedule* for the game like this (our notation here is inspired by

Pudlák's work on matches made up of several games [31]):

$$\begin{tikzcd}
H: & D & C \arrow[d] & D & \dots \arrow[d, phantom, "\cdots"] & C \arrow[d] \\
G: & C \arrow[u] & D & C \arrow[u] & \dots & D
\end{tikzcd}$$

First C plays at the bottom of the first column, as the first move in G; then D plays at the top of the first column, as the first move in H; then C plays at the top of the second column, as the second move in H; then D plays at the bottom of the second column, as the second move in G; and so on.

D wins this game if either all the moves played by both players in the top row satisfy $\neg H$ (i.e. constitute a win for Player A in H), or all the moves played in the bottom row satisfy G (i.e. constitute a win for Player B in G). In other words, D wins this reduction game if he wins either the top or the bottom row. Hence a polynomial time winning strategy for D is the same thing as a reduction of G to H, and to show that one game is reducible to another we will usually do it by describing such a strategy.

We can now prove Theorem 4, that $1{-}\mathrm{Ref}(\mathrm{PK}_k^0)$ is reducible to GI_{k+2}. We give a proof that can be formalized in PV.

Proof of Theorem 4 Let a be the total length of the refutation and let b be an upper bound on the total number of formulas that appear in it (recall that the structure of the formulas is completely given by Q). Necessarily a and b are quasipolynomial in the size parameter of our instance of $1{-}\mathrm{Ref}(\mathrm{PK}_k^0)$. We will think of the refutation as consisting of a sequence $C_1, \dots, C_a$ of cedents, where C_a is the empty cedent.

We will uniformly define games $G_1, \dots, G_a$. The moves will be numbers $< b$, representing formulas. The idea of the game G_i is that in the first turn Player A names a cedent C_x with $x \leq i$; in the second turn Player B names a formula y that appears in C_x and claims that y is true (in the assignment α); and the remaining turns consist of a back-and-forth between the players, with A trying to show that y is false and B trying to maintain that it is true. Player B can always win G_i for small i, since it is easy to find a true formula in an initial clause; Player A can always win G_a because it is impossible to find a true formula in the final, empty cedent; and the reduction between G_{i+1} and G_i comes from soundness and the form of the rules.

To define game G_i formally, we describe what the players have to do at each turn. If at any turn one fails to do this then the other player immediately wins. We state in brackets the informal claim that a player makes about his turn. These claims have high quantifier complexity and the following turns of the game represent checking their truth value.

After turn 2, the game can continue in two different ways, depending on whether y is a conjunction or a disjunction. We describe the case for a conjunction first.

1. A names a cedent C_x with $x \leq i$ (claiming it contains no true formulas)

2. B names a formula $y \in C_x$ (claiming it is true)

3. A names a disjunction $z_1 \in y$ (claiming it is false)
4. B names a conjunction $z_2 \in z_1$ (claiming it is true)

   etc.

The game continues in this way, with the level of the named formula dropping by one each turn, until one of the players names a single (unpadded) literal. Then B wins if this literal is true and A wins if it is false. Note that the game must have finished after $k+2$ turns.

If the first formula y named by B is a disjunction, then the game proceeds like this:

3. A misses his turn (waiting for B to substantiate his claim that y is true by naming a true disjunct)
4. B names a conjunction $z_1 \in y$ (claiming it is true)

then the game continues as before. We include this missed turn, rather than just combining B's turns 2 and 4 into one turn, because the way the turns in successive games match up with each other will be important for our reductions between games. Note that again the game must have finished after $k+2$ turns, since we only allow disjunctions of level $k-1$ or lower to appear in a PK_k^0 proof.

We must now give the strategies $U, V, W_1, \ldots, W_{a-1}$ and show that, from any witness that condition 1, 2 or 3 of the definition of game induction is false, we can derive in polynomial time a witness to our instance of $1{-}\mathrm{Ref}(\mathrm{PK}_k^0)$.

U is the following strategy for Player B in G_1. Player A can only name the initial cedent C_1 in his first turn (otherwise he loses immediately). B then looks at the first clause of the CNF F. Recall that this is narrow, that is, contains only polynomially many literals (in the length of the size parameter) and that these are listed by a polynomial time function. B goes through these literals and plays the first true such literal (under the assignment α) as his move y. If there is no such literal, he surrenders the game by playing an illegal move.

Suppose that this strategy is unsuccessful. Then either there is no true literal in the first clause of F, in which case the first clause of F is falsified by α, yielding a witness to condition 4 of the definition of $1{-}\mathrm{Ref}(\mathrm{PK}_k^0)$; or B played a true literal y from the first clause of F but lost because y is not a member of C_1, in which case we have a witness to condition 3 of the definition of $1{-}\mathrm{Ref}(\mathrm{PK}_k^0)$, since C_1 is just the initial cedent R_1.

V is the following strategy for Player A in G_a. A plays the final cedent C_a of the refutation. If B is able to play anything successfully in the next turn, then B's move must witness that C_a is not the empty cedent and that the refutation is not well-formed.

The reduction W_i of G_{i+1} to G_i is more complicated. We will describe it in the form of a strategy for D in a reduction game against C. When we talk about C or D playing in the "top row" or "bottom row" we are referring to the schedule of the reduction game. We suppose that C has named a cedent $x \leq i+1$ in his first turn. Then D's strategy will depend on the rule by which C_x was derived.

Case 0: C_x is an initial clause. In this case D ignores the top row of the reduction game and plays on the bottom row as B does in the strategy U.

Case 1: C_x is derived from some $C_{x'}$ by padding introduction,

$$\frac{C_{x'} : \Gamma, p}{C_x : \Gamma, (p)}.$$

D's first move is to play x' in the top row, to which C replies with some $y \in C_{x'}$. Now D's strategy will depend on whether p is true or false. This is checkable in polynomial time, since p is just a single literal.

Suppose p is true. Then D ignores the top row from now on and plays "(p)" in the bottom row. We use quotation marks here because the players cannot know at this point whether the formula picked out by the function f to play the role of "(p)" actually has the form that (p) should have, since this property is $\hat{\Pi}^b_1$. However, by the definition of the inference rule, f should provide a sequence of names of formulas "(p)"$= \phi_1, \phi_2, \dots, \phi_l = p$ where each formula is the unique element of the previous formula, since otherwise the proof is not well-formed. So if C responds (on the bottom row) with an element of (p) which is not ϕ_2 then we have a witness that the proof is not well-formed. Otherwise D responds to ϕ_2 with ϕ_3; and so on. Eventually either we get a witness to $1-\mathrm{Ref}(\mathrm{PK}^0_k)$ or D wins the bottom row.

If p is false, then C's reply y must be a side formula from Γ (that is, some non-auxiliary formula), since he loses the top immediately if he replies with p. From this point, D can copy C's moves in every column (in the schedule) and will thus win either the top or the bottom. The only way this can fail is if C played a side formula on the top which does not exist on the bottom, in which case we have a witness that the proof is not well-formed.

Case 2: Padding elimination,

$$\frac{C_{x'} : \Gamma, (p)}{C_x : \Gamma, p}.$$

Similar to case 1.

Case 3: Internal $\bigwedge$-introduction,

$$\frac{C_{x'} : \Gamma, \Phi[\bigwedge X] \qquad\qquad C_{x''} : \Gamma, p}{C_x : \Gamma, \Phi[\bigwedge X \wedge (p)]}.$$

If p is false, then D plays x'' at the top and then copies C's moves, as above. If p is true, then the interesting case of D's strategy looks like this:

$$\begin{tikzcd}
 & \bot & \top & & \top & \bot & \\
G_i : & x' & \Phi[\bigwedge X] \arrow[d] & \dots & \bigwedge X \arrow[d] & z & \dots \\
G_{i+1} : & x \arrow[u] & \Phi[\bigwedge X \wedge (p)] & \dots & \bigwedge X \wedge (p) & z \in X \arrow[u] & \dots
\end{tikzcd}$$

We have marked each column with $\bot$ or $\top$ to remind the reader what claims the players are making in those turns.

We will describe this strategy in detail. C claims that the cedent C_x contains no true formula. D responds that, in that case, the cedent $C_{x'}$ contains no true formula. C replies that, in fact, the auxiliary formula "$\Phi[\bigwedge X]$" is true in $C_{x'}$ (alternatively C replies that some side formula is true; but then either the proof is not well-formed, or D can copy C's moves from now on and thus win). D says that this means that the principal formula "$\Phi[\bigwedge X \wedge (p)]$" is true in C_x.

Now f should list sequences "$\Phi[\bigwedge X]$"$= \phi_1, \phi_2, \ldots, \phi_l =$"$\bigwedge X$" and "$\Phi[\bigwedge X \wedge (p)]$"$= \phi'_1, \phi'_2, \ldots, \phi'_l =$"$\bigwedge X \wedge (p)$" through these formulas to the relevant subformulas. As long as C sticks to naming formulas from these paths (ϕ_js on the top and ϕ'_js on the bottom), D replies with the corresponding formula from the other side. These paths should lead to the only place where the auxiliary formula and the principal formula differ. So if C names a subformula off the path, then either D can copy C's moves from that point or the proof is not well-formed.

In the last column shown, C must name a member z of $\bigwedge X \wedge (p)$, claiming it is false. He cannot name "(p)", because then D could go on to win at the bottom as in case 1, since p is true. But if C names any $z \in X$, then D can reply with the same z, as shown (otherwise z is a witness that the proof is not well-formed), and go on to copy C's moves from that point.

Note that if $\Phi[\bigwedge X]$ and $\Phi[\bigwedge X \wedge (p)]$ are disjunctions rather than conjunctions then, by the rules of G_i and G_{i+1}, both players skip their turns in the third column. Otherwise the strategy is the same.

Case 4: Internal $\bigvee$-elimination,

$$\frac{C_{x'} : \Gamma, \Phi[\bigvee X \vee (p)]}{C_x : \Gamma, \Phi[\bigvee X], p}.$$

If p is true then D can win the bottom by playing p. If p is false, then the interesting case of D's strategy looks like this

$$\begin{array}{rcccccc}
 & \bot & \top & & \bot & \top & \\
G_i: & x' & \Phi[\bigvee X \vee (p)] & \ldots & \bigvee X \vee (p) & z \in X & \ldots \\
 & \uparrow & \downarrow & & \uparrow & \downarrow & \\
G_{i+1}: & x & \Phi[\bigvee X] & \ldots & \bigvee X & z & \ldots
\end{array}$$

and is similar to the previous case. If C plays any other move before the last column shown, then D can go on to win by copying C's moves. If C had played (p) in the last column shown, then C would be claiming that (p) is true and D could go on to win the top.

Case 5: Internal weakening,

$$\frac{C_{x'} : \Gamma, \Phi[\bigvee X]}{C_x : \Gamma, \Phi[\bigvee Y]}$$

where $X \subseteq Y$. D's strategy looks like:

$$\begin{array}{rcccccc}
 & \bot & & \bot & \top & \\
G_i : & x' & \dots & \bigvee X & z \in X & \dots \\
 & \uparrow & & \uparrow & \downarrow & \\
G_{i+1} : & x & \dots & \bigvee Y & z & \dots
\end{array}$$

Case 6: Internal $\bigwedge$-elimination,

$$\frac{C_{x'} : \Gamma, \Phi[\bigwedge X]}{C_x : \Gamma, \Phi[\bigwedge Y]}$$

where $X \supseteq Y$. D's strategy is dual to case 5.

Case 7: Weakening,

$$\frac{C_{x'} : \Gamma}{C_x : \Delta}$$

where $\Gamma \subseteq \Delta$. Trivial.

Case 8: $\bigwedge$-introduction,

$$\frac{C_{x'} : \Gamma, F}{C_x : \Gamma, \bigwedge\{F\}}$$

where by the rules about levels, F must be a disjunction, say $\bigvee X$. D's strategy looks like:

$$\begin{array}{rcccccc}
 & \bot & \top & \bot & \top & \\
G_i : & x' & \bigvee X & \text{skip} & z \in X & \dots \\
 & \uparrow & \downarrow & \uparrow & \downarrow & \\
G_{i+1} : & x & \bigwedge\{\bigvee X\} & \bigvee X & z & \dots
\end{array}$$

Case 9: $\bigwedge$-elimination,

$$\frac{C_{x'} : \Gamma, \bigwedge\{F\}}{C_x : \Gamma, F}.$$

Again F must be a disjunction $\bigvee X$. D's strategy looks like:

$$\begin{array}{rcccccc}
 & \bot & \top & \bot & \top & \\
G_i : & x' & \bigwedge\{\bigvee X\} & \bigvee X & z \in X & \dots \\
 & \uparrow & \downarrow & \uparrow & \downarrow & \\
G_{i+1} : & x & \bigvee X & \text{skip} & z & \dots
\end{array}$$

Case 10: Resolution,

$$\frac{C_{x'} : \Gamma, p \qquad\qquad C_{x''} : \Gamma, \neg p}{C_x : \Gamma}.$$

If p is false then D plays x' in the first column, otherwise he plays x''. The rest of the strategy is trivial. $\square$

4 Translation into a polynomial time refutation

In this section we show that the $\forall\hat{\Sigma}_1^b$ consequences of T_2^{k+2} are reducible to $1-\mathrm{Ref}(\mathrm{PK}_k^0)$. We first show how to translate a T_2^{k+2} proof into a polynomial time PK_k refutation. Then we show how this translation can be changed to give a polynomial time PK_k^0 refutation, by simulating the usual rules for connectives using the rules of PK_k^0. Throughout, k is a fixed natural number.

Just for this section, we define $\tilde{\Sigma}_i^b$ and $\tilde{\Pi}_i^b$ formulas to be formulas of precisely the complexity $\hat{\Sigma}_i^b$ or $\hat{\Pi}_i^b$ and not of lower complexity, unless we say so explicitly, where for example "of complexity $\tilde{\Pi}_i^b$ or lower" will mean $\tilde{\Pi}_j^b$ for $j \leq i$ or $\tilde{\Sigma}_j^b$ for $j < i$. Furthermore the quantifiers in such a formula must alternate; we do not allow blocks of the same quantifier, repeated. Finally the terms appearing as the bounds on quantifiers can only contain free variables, not bound variables.

Definition 16 *Given a $\tilde{\Pi}_i^b$ formula ϕ and an assignment $\bar{b}$ to the free variables in ϕ, the propositional translation $\langle\phi\rangle_{\bar{b}}$ is formally a tuple recording the complexity of ϕ (i.e. "$\tilde{\Pi}_i^b$"), the bounds on all the quantifiers (evaluated using $\bar{b}$), the Gödel number of the sharply bounded part θ of ϕ, and the parameters from $\bar{b}$ corresponding to the free variables of θ. We think of the quantifiers as having been translated into connectives and the sharply bounded matrix as having been translated into a family of propositional variables, one for each choice of parameters arising from the connectives. Similarly for $\tilde{\Sigma}_i^b$ formulas. We will normally abuse notation and either not write the parameters at all or write the formula as $\langle\phi(\bar{b})\rangle$ rather than $\langle\phi\rangle_{\bar{b}}$.*

We call a propositional formula full *if it arises in this way, as a translation of a bounded first order formula.*

For $d \in \mathbb{N}$ we write Q_t^d for the polynomial time k-formula table that encodes the structure of all full propositional formulas arising from first order formulas of complexity $\tilde{\Pi}_k^b$ or lower and with sharply bounded parts smaller than d and parameters and bounds smaller than t.

For a formula σ of complexity $\tilde{\Pi}_{k+2}^b$ or lower, a term t and a natural number d, we say that t and d *dominate* σ if d is bigger than the Gödel number of the sharply bounded part of σ and t dominates the bounds on all quantifiers in σ.

Given such a σ, t and d and a set of parameters for the free variables in σ, we will translate σ into a *polynomial time table of cedents* σ°. Formally this is a polynomial time binary relation defining a $r \times |Q_t^d|$ table which takes the parameters as an extra set of arguments. The height r is given as a polynomial time function of the parameters. Each column represents a formula from Q_t^d and each row represents a cedent. The original first order formula will be true if and only if in every row at least one formula is true.

It is easy to replace such a table with one with a larger d or t, representing having a larger set of formulas to draw upon. We do this implicitly in the constructions in this section.

We will usually write a singleton cedent of the form $\{\langle\sigma\rangle\}$ just as $\langle\sigma\rangle$.

Definition 17 *If σ is $\tilde{\Pi}_k^b$ or of lower complexity then σ° is the single, singleton cedent $\langle\sigma\rangle$.*

If σ is $\tilde{\Sigma}^b_{k+1}$ or $\tilde{\Sigma}^b_k$, and has the form $\exists x\!<\!r\,\phi(x)$, then σ° is the single cedent $\{\langle\phi(i)\rangle : i < r\}$.

If σ is $\tilde{\Pi}^b_{k+2}$ or $\tilde{\Pi}^b_{k+1}$, and has the form $\forall x\!<\!r\,\exists y\!<\!s\,\phi(x,y)$, then σ° is the table of cedents $(\{\langle\phi(i,j)\rangle : j < s\})_{i<r}$.

If $k = 0$ and σ is $\tilde{\Pi}^b_1$, and has the form $\forall x\!<\!r\,\phi(x)$, then σ° is the table of singleton cedents $(\langle\phi(i)\rangle)_{i<r}$.

Definition 18 *Let A and B be polynomial time tables of cedents, with r and s rows respectively. Then $A + B$ is the $r + s$ row table made by concatenating A and B. $A * B$ is the rs row table whose rows are the unions of pairs of rows from A and B. Precisely, the $(s(i-1)+j)$th row of $A * B$ is the union of the ith row of A and the jth row of B. Note that these operations are associative and that these new tables are still polynomial time.*

The translation Γ° of a first order cedent $\Gamma = (\gamma_1, \ldots, \gamma_m)$ is the table $\gamma_1^\circ + \ldots + \gamma_m^\circ$. The translation Γ^* is the table $\gamma_1^\circ * \ldots * \gamma_m^\circ$. A first order cedent only contains finitely many formulas, so these tables are still polynomial time.

Our translation from first order into propositional proofs is essentially that of Paris and Wilkie [27]. Krajíček showed using this translation that T^{k+2}_2 proofs could in fact be translated into depth $k+1/2$, quasipolynomial size propositional proofs [17, 19]. In [22] it was shown that these proofs can have polynomial time structure (in our sense) for $k = 0$ or 1. The construction in this section is a generalization of this last result.

The next proof constructs a polynomial time PK_k derivation. This is defined similarly to a polynomial time PK^0_k derivation, except that the rules allowed are: weakening; resolution; $\bigwedge$-introduction, i.e. if ϕ is $\tilde{\Sigma}^b_j$ for $j < k$ then from the table $(\Gamma \cup \langle\phi(i)\rangle)_{i<r}$ we can derive the cedent $\Gamma \cup \langle\forall x\!<\!r\,\phi(x)\rangle$; $\bigwedge$-elimination, i.e. the reverse of this; $\bigvee$-introduction, i.e. if ϕ is $\tilde{\Pi}^b_j$ for $j < k-1$ then from the cedent $\Gamma \cup \{\langle\phi(i)\rangle : i < r\}$ we can derive the cedent $\Gamma \cup \langle\exists x\!<\!r\,\phi(x)\rangle$; and $\bigvee$-elimination, i.e. the reverse of this.

We follow [9] in our formulation of the first order sequent calculus for bounded arithmetic.

Theorem 19 *Suppose Π is a proof in the sequent calculus for T^{k+2}_2 in which every formula is of complexity $\tilde{\Pi}^b_{k+2}$ or lower, and which ends with the sequent $\Gamma \longrightarrow \Delta$.*

Then there exist a constant d and a term t, dominating every formula in Π, and a polynomial time table A of auxiliary clauses, such that there is a polynomial time PK_k derivation of Δ^* from $\Gamma^\circ + A$ where all formulas in the derivation come from the table Q^d_t. Here the auxiliary clauses A are cedents containing only literals; they are narrow, meaning that the table comes with a polynomial time function listing the polynomially many literals in each clause; and every clause is true, under the natural truth assignment to literals.

Everything takes as parameters an assignment to the free variables in Γ and Δ. The statement that the derivation and the auxiliary clauses are well-formed, for all choices of parameters, is provable in PV.

Proof The proof is by induction on the number of steps in Π. The induction step splits into cases depending on the way in which the final sequent in Π is derived. In each case the new bounding term t and constant d are straightforward and we will not give details.

BASIC axioms, equality axioms, propositional rules, sharply bounded quantifier rules
Every formula in Π is $\tilde{\Pi}^b_{k+2}$ and in particular is strict, with propositional connectives and sharply bounded quantifiers only occurring in the sharply bounded part of a formula. So the associated rules are only applied to sharply bounded formulas, corresponding to level 0 formulas in our translation. Furthermore, instances of the BASIC and equality axioms translate directly into level 0 formulas. Hence all these things can be taken care of by adding an appropriate table of auxiliary clauses and some resolution steps.

We give one example (the same one as in [22]), for the right sharply bounded universal quantifier introduction rule, which has the form:

$$\frac{x < |s|, \Gamma \longrightarrow \Delta, \phi(x)}{\Gamma \longrightarrow \Delta, \forall y {<} |s| \, \phi(y)}$$

where our assumption tells us that ϕ must be sharply bounded.

By the inductive hypothesis we have, uniformly in an assignment r to x (and assignments to any other free variables), a derivation π_r of $\Delta^* * \langle \phi(r) \rangle$ from $\langle r < |s| \rangle + \Gamma^\circ + A_r$ where A_r is a table of auxiliary clauses. Let s' be the value of $|s| - 1$ and let A be the table of auxiliary clauses $A_0 + \ldots + A_{s'} + \langle 0 < |s| \rangle + \ldots + \langle s' < |s| \rangle + \{\neg\langle\phi(0)\rangle, \ldots, \neg\langle\phi(s')\rangle, \langle \forall y {<} |s| \, \phi(y) \rangle\}$ (first padding out all tables $A_0, \ldots, A_{s'}$ to the same length so that their concatenation has a uniform internal structure).

Then for each $r < |s|$ using π_r we can derive $\Delta^* * \langle \phi(r) \rangle$ from $\Gamma^\circ + A$. Then we can derive each cedent in $\Delta^* * \langle \forall y {<} |s| \, \phi(y) \rangle$ by resolving the last clause in A with the corresponding cedent in $\Delta^* * \langle \phi(r) \rangle$ for every r in turn.

Weak structural rules
The exchange and contraction rules just require easy changes of the cedents pointed to by the function f.

Left weakening is trivial. Right weakening

$$\frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \phi}$$

requires the introduction of some new weakening steps at the end of our propositional derivation, one for each pair of a cedent from Δ^* and a cedent from ϕ°.

Cut rule
Suppose the last inference in Π has the form

$$\frac{\Gamma \longrightarrow \Delta, \phi \qquad \phi, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}.$$

By the inductive hypothesis, we have uniform derivations π_L, π_R and uniform tables of auxiliary clauses A_L, A_R for respectively the left and right upper sequents. We can write these as:

$$\pi_L : \Gamma^\circ + A_L \vdash \Delta^* * \phi^\circ \qquad \text{and} \qquad \pi_R : \phi^\circ + \Gamma^\circ + A_R \vdash \Delta^*.$$

A polynomial time proof is a table of cedents (with some extra structure given by the function f), so it makes sense to use our operator $*$ on proofs and to write $\Delta^* * \pi_R$ for the object obtained by taking, for each cedent C in Δ^*, a copy of π_R with C added (by union) to every cedent. Allowing for some simple changes to this object to add initial and final clauses in the right places and to extend f appropriately, we have

$$\Delta^* * \pi_R : \Delta^* * \phi^\circ + \Delta^* * \Gamma^\circ + \Delta^* * A_R \vdash \Delta^* * \Delta^*$$

which, if we tidy up the conclusion and add some weakening steps to derive $\Delta^* * \Gamma^\circ$ from Γ° and $\Delta^* * A_R$ from A_R, yields a polynomial time derivation

$$\Delta^* * \phi^\circ + \Gamma^\circ + A_R \vdash \Delta^*.$$

Finally, by preceding this with π_L we can form a polynomial time derivation

$$\Gamma^\circ + A_L + A_R \vdash \Delta^*$$

as required.

Induction rule
We may assume that the induction rule occurs in the form

$$\frac{\Gamma, \phi(s), r = s+1 \longrightarrow \phi(r), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(w), \Delta}.$$

This is because we can derive the normal induction rule

$$\frac{\Gamma, \phi(s) \longrightarrow \phi(s+1), \Delta}{\Gamma, \phi(0) \longrightarrow \phi(w), \Delta}$$

from our new rule without increasing the quantifier complexity of the proof, by first deriving $r = s+1, \phi(s+1) \longrightarrow \phi(r)$ using equality and quantifier axioms, then cutting with the upper sequent of the normal induction rule, then applying our new induction rule.

The treatment of it is similar to that of the cut rule. By the inductive hypothesis, for the upper sequent for each $s < w$ we have derivations π_s and tables of auxiliary clauses A_s, uniform in s, with

$$\pi_s : \Gamma^\circ + \phi(s)^\circ + A_s \vdash \phi(s+1)^\circ * \Delta^*.$$

By similar manipulations to those for cut, we can obtain derivations π'_s with

$$\pi'_s : \Gamma^\circ + A_s + \phi(s)^\circ * \Delta^* \vdash \phi(s+1)^\circ * \Delta^*.$$

Finally by padding all these proofs out to the same length, concatenating them and connecting them by extending the function f, we get a polynomial time derivation

$$\Gamma^\circ + A + \phi(0)^\circ * \Delta^* \vdash \phi(w)^\circ * \Delta^*,$$

where A is the concatenation of all the tables of auxiliary clauses A_s. Adding a few more weakening steps at the start gives us the required derivation.

Bounded left $\exists$ introduction
Suppose the last inference in Π has the form

$$\frac{x < s, \phi(x), \Gamma \longrightarrow \Delta}{\exists y {<} s\, \phi(y), \Gamma \longrightarrow \Delta}.$$

We first consider the case when ϕ is a $\tilde{\Pi}^b_k$ or $\tilde{\Pi}^b_{k-1}$ formula. For each r we have

$$\pi_r : (r < s)^\circ + \phi(r)^\circ + \Gamma^\circ + A_r \vdash \Delta^*.$$

For $r < s$ we can let A'_r be A_r with $\{\langle r < s\rangle\}$ added as an extra auxiliary clause, giving

$$\pi'_r : \phi(r)^\circ + \Gamma^\circ + A'_r \vdash \Delta^*.$$

Now by a similar manipulation to the case of cut, by premultiplying the proof by the table $\Delta^* * (\exists y < r\, \phi(y))^\circ$ and then doing some tidying up and some weakening, we can obtain a derivation

$$\Delta^* * (\exists y {<} r\, \phi(y))^\circ * \phi(r)^\circ + \Gamma^\circ + A'_r \vdash \Delta^* * (\exists y {<} r\, \phi(y))^\circ.$$

But by the definition of our translation, $(\exists y < r\, \phi(y))^\circ * \phi(r)^\circ$ is precisely the cedent $(\exists y {<} r{+}1\, \phi(y))^\circ$. So for each $x < r$ we have a polynomial time derivation

$$\pi''_r : \Delta^* * (\exists y {<} r + 1\, \phi(y))^\circ + \Gamma^\circ + A'_r \vdash \Delta^* * (\exists y {<} r\, \phi(y))^\circ.$$

By padding all the proofs $\pi''_{s-1}, \ldots, \pi''_0$ out to the same length, concatenating them, extending f suitably, observing that $(\exists y {<} 0\, \phi(y))^\circ$ is the empty sequent, and adding some extra weakening at the start to obtain $\Delta^* * (\exists y {<} s\, \phi(y))^\circ$ from $(\exists y {<} s\, \phi(y))^\circ$, we get the required derivation

$$(\exists y {<} s\, \phi(y))^\circ + \Gamma^\circ + A \vdash \Delta^*.$$

In the case when ϕ is $\tilde{\Pi}^b_{k-2}$ or lower we do the same construction as above, simply replacing cedents of the form $(\exists y < r\, \phi(y))^\circ$ with the (in this case, different) cedent $\{\langle \phi(i)\rangle : i < r\}$. This gives a derivation

$$\{\langle \phi(i)\rangle : i < s\} + \Gamma^\circ + A \vdash \Delta^*$$

and we can obtain the desired derivation by adding an $\bigvee$-elimination step at the start to derive $\{\langle \phi(i)\rangle : i < r\}$ from $(\exists y < r\, \phi(y))^\circ$ (which, in this case, is just the cedent $\{\langle \exists y {<} r\, \phi(y)\rangle\}$).

Bounded left $\forall$ introduction
Suppose the last inference in Π has the form

$$\frac{\phi(r), \Gamma \longrightarrow \Delta}{r < s, \forall x{<}s\, \phi(x), \Gamma \longrightarrow \Delta}.$$

We may assume $r < s$. Otherwise we do not need the inductive hypothesis since we can derive the empty clause from $(r < s)^\circ$ if we add $\{\neg\langle r < s\rangle\}$ as an auxiliary clause, so we can easily get a derivation for the bottom sequent using weakening.

We first consider the case when ϕ is $\tilde{\Sigma}^b_{k+1}$ or $\tilde{\Sigma}^b_k$. In this case $(\forall x{<}s\, \phi(x))^\circ$ is a table of cedents. This table already includes the cedent $\phi(r)^\circ$ and it is easy to form the derivation for the bottom sequent from the derivation for the top sequent.

If ϕ if $\tilde{\Sigma}^b_{k-1}$ or below then we use a similar construction, with the difference that at the beginning we need an $\bigwedge$-elimination step to derive a table of cedents $(\{\langle\phi(j)\rangle\})_{j<s}$ from $(\forall x{<}s\, \phi(x))^\circ$.

Bounded right $\exists$ introduction
Suppose the last inference in Π has the form

$$\frac{\Gamma \longrightarrow \Delta, \phi(r)}{r < s, \Gamma \longrightarrow \Delta, \exists x{<}s\, \phi(x)}.$$

As above we may assume $r < s$.

The construction is to take the derivation for the top sequent, and then add to the end a derivation of $\Delta^* * (\exists x{<}s\, \phi(x))^\circ$ from $\Delta^* * \phi(r)^\circ$. Similarly to the earlier cases, if ϕ is $\tilde{\Pi}^b_k$ or $\tilde{\Pi}^b_{k-1}$ then this just needs some weakening steps; if ϕ is of lower complexity then it needs some weakening and some $\bigvee$-introduction steps.

Bounded right $\forall$ introduction
Suppose the last inference in Π has the form

$$\frac{x < s, \Gamma \longrightarrow \Delta, \phi(x)}{\Gamma \longrightarrow \Delta, \forall y{<}s\, \phi(y)}$$

Using the inductive hypothesis we can make a derivation which starts with Γ° and derives $\Delta^* * \phi(r)^\circ$ for each $r < s$. If ϕ is $\tilde{\Sigma}^b_k$ or $\tilde{\Sigma}^b_{k+1}$ then this is exactly the derivation required. Otherwise, we need to add some $\bigwedge$-introduction steps. $\Box$

All propositional formulas appearing so far are full. For our PK^0_k proof, however, we will need formulas from a more general class.

Definition 20 *Level* 0 almost-full *formulas are the same as level* 0 *full formulas. For each $i < k$, a level $i + 1$ almost-full conjunction A is specified by a level $i + 1$ full conjunction B and a level i almost-full disjunction C; the set of conjuncts of A is $\{X : X$ is a conjunct of $B\} \cup \{C\}$. Almost-full disjunctions are defined similarly. As in the case of full formulas, the almost-full formulas can be given by a polynomial time k-formula table with a bound d on complexity and a bound t on the size of the parameters.*

Theorem 21 *Theorem 19 also holds for the system* PK_k^0.

Proof We will show that we can derive PK_k's rules for connectives using polynomial time PK_k^0 derivations made up of almost-full formulas.

The proof is by induction on the level of a formula. The hypothesis is that for every almost-full level i formula F, and every almost-full formula Φ with a full subformula (that is, a node in the tree for Φ) Z of level $i+1$ or higher, we can derive the following generalizations of rules 1–4 of PK_k^0 by polynomial time PK_k^0 derivations (using the parameters that define the formulas):

$$\frac{F}{(F)} \text{ and } \frac{(F)}{F} \qquad \text{where padding can be to any level;}$$

$$\frac{\Phi[Z] \qquad F}{\Phi[Z \wedge (F)]} \qquad \text{if } Z \text{ is a conjunction;}$$

$$\frac{\Phi[Z \vee (F)]}{\Phi[Z], F} \qquad \text{if } Z \text{ is a disjunction.}$$

Side formulas are allowed but we will not write them here. We will also leave out weakening steps. Our hypothesis is true for level 0. Suppose it is true for level i.

Padding introduction and elimination
The construction for padding introduction is similar to the one given below for internal $\bigwedge$-introduction. Similarly for padding elimination and internal $\bigvee$-elimination.

Internal $\bigwedge$-introduction
We will derive the rule

$$\frac{\Phi[\bigwedge X] \qquad F}{\Phi[\bigwedge X \wedge (F)]}$$

for $\bigwedge X$ a full formula. First suppose F is a level $i+1$ conjunction $\bigwedge\{G_1, \ldots, G_n, H\}$ of level i disjunctions, where $\bigwedge\{G_1, \ldots, G_n\}$ is full. Our derivation is

$\Phi[\bigwedge X]$	
$\bigwedge\{G_1, \ldots, G_n, H\}$	
$\bigwedge\{G_1\}$	internal $\bigwedge$-elimination
G_1	$\bigwedge$-elimination
$\vdots$	
H	as for G_1
$\Phi[\bigwedge X \wedge (\bigwedge\{G_1\})]$	level i internal $\bigwedge$-introduction (ind. hyp.)
$\Phi[\bigwedge X \wedge (\bigwedge\{G_1, G_2\})]$	level i internal $\bigwedge$-introduction
$\vdots$	
$\Phi[\bigwedge X \wedge (\bigwedge\{G_1, \ldots, G_n, H\})]$.	

Notice that the level i internal $\bigwedge$-introduction steps, except for the first one, are of the form

$$\frac{\Phi'[\bigwedge\{G_1,\ldots,G_m\}] \qquad\qquad I}{\Phi'[\bigwedge\{G_1,\ldots,G_m,I\}]}$$

for some $m \leq n$. We have derivations for these by the inductive hypothesis and the fact that $\bigwedge\{G_1,\ldots,G_m\}$ is full.

Now suppose F is a level $i+1$ disjunction $\bigvee\{G_1,\ldots,G_n,H\}$ of level i conjunctions, where $\bigvee\{G_1,\ldots,G_n\}$ is full. Our derivation is

$\Phi[\bigwedge X]$	
$\bigvee\{G_1,\ldots,G_n,H\}$	
$\bigvee\{G_1,\ldots,G_n\},H$	level i internal $\bigvee$-elimination
$\vdots$	
$\bigvee\{G_1\},G_2,\ldots,G_n,H$	
$G_1,\ldots,G_n,H$	level i padding elimination
$\Phi[\bigwedge X\wedge(\bigvee\{G_1\})],G_2,\ldots,G_n,H$	level i internal $\bigwedge$-introduction
$\Phi[\bigwedge X\wedge(F)],G_2,\ldots,G_n,H$	internal weakening
$\Phi[\bigwedge X\wedge(F)],\Phi[\bigwedge X\wedge(\bigvee\{G_2\})],$ $G_3,\ldots,G_n,H$	level i internal $\bigwedge$-introduction
$\Phi[\bigwedge X\wedge(F)],G_3,\ldots,G_n,H$	internal weakening
$\vdots$	
$\Phi[\bigwedge X\wedge(F)]$	

Internal $\bigvee$-elimination
We will derive the rule

$$\frac{\Phi[\bigvee X\vee(F)]}{\Phi[\bigvee X],F}$$

for $\bigvee X$ a full formula. First suppose F is a level $i+1$ conjunction $\bigwedge\{G_1,\ldots,G_n,H\}$ of level i disjunctions, where $\bigwedge\{G_1,\ldots,G_n\}$ is full. Our derivation is

$\Phi[\bigvee X\vee(\bigwedge\{G_1,\ldots,G_n,H\})]$	
$\Phi[\bigvee X\vee(\bigwedge\{G_1\})]$	internal $\bigwedge$-elimination
$\Phi[\bigvee X],G_1$	level i internal $\bigvee$-elimination
$\vdots$	
$\Phi[\bigvee X],H$	as for G_1
$\Phi[\bigvee X],\bigwedge\{G_1\}$	$\bigwedge$-introduction
$\Phi[\bigvee X],\bigwedge\{G_1,G_2\}$	level i internal $\bigwedge$-introduction
$\vdots$	
$\Phi[\bigvee X],\bigwedge\{G_1,\ldots,G_n,H\}$	

Now suppose F is a level $i+1$ disjunction $\bigvee\{G_1,\ldots,G_n,H\}$ of level i conjunctions, where $\bigvee\{G_1, \ldots, G_n\}$ is full. Our derivation is

$\Phi[\bigvee X \vee (\bigvee\{G_1, \ldots, G_n, H\})]$	
$\Phi[\bigvee X \vee (\bigvee\{G_1, \ldots, G_n\})], H$	level i internal $\bigvee$-elimination
$\vdots$	
$\Phi[\bigvee X \vee (\bigvee\{G_1\})], G_2, \ldots, G_n, H$	
$\Phi[\bigvee X], G_1, \ldots, G_n, H$	level i internal $\bigvee$-elimination
$\Phi[\bigvee X], \bigvee\{G_1\}, G_2, \ldots, G_n, H$	level i padding introduction
$\Phi[\bigvee X], \bigvee\{G_1, \ldots, G_n, H\}, G_2, \ldots, G_n, H$	internal weakening
$\Phi[\bigvee X], \bigvee\{G_1, \ldots, G_n, H\}, \bigvee\{G_2\}, G_3, \ldots, G_n, H$	level i padding introduction
$\Phi[\bigvee X], \bigvee\{G_1, \ldots, G_n, H\}, G_3, \ldots, G_n, H$	internal weakening
$\vdots$	
$\Phi[\bigvee X], \bigvee\{G_1, \ldots, G_n, H\}$	

This concludes the inductive construction of our derivations.

Now we can easily derive the rules of PK_k. For example $\bigvee$-introduction is

$G_1, \ldots, G_n$	
$\bigvee\{G_1\}, G_2, \ldots, G_n$	padding introduction
$\bigvee\{G_1, \ldots, G_n\}, G_2, \ldots, G_n$	internal weakening
$\bigvee\{G_1, \ldots, G_n\}, \bigvee\{G_2\}, G_3, \ldots, G_n$	padding introduction
$\bigvee\{G_1, \ldots, G_n\}, G_3, \ldots, G_n$	internal weakening
$\vdots$	
$\bigvee\{G_1, \ldots, G_n\}$	

□

We can now prove Theorem 5 that, for $k \geq 0$, $\forall\hat{\Sigma}_1^b(T_2^{k+2})$ is reducible to $1\mathrm{-Ref}(\mathrm{PK}_k^0)$ by reductions that work provably in PV.

Proof of Theorem 5 Suppose T_2^{k+2} proves $\forall x\, \exists y < x\, \phi(x, y)$ where ϕ is sharply bounded. We may assume that the calculus for T_2^{k+2} uses the induction rule only for $\tilde{\Pi}_{k+2}^b$ or lower formulas. Hence by free-cut elimination (see for example [9]; note that our class of formulas is closed under taking subformulas and freely substituting terms for variables) there is a first order derivation, in our calculus for T_2^{k+2}, of the sequent $\forall y < x\, \neg\phi(x, y) \longrightarrow \emptyset$ in which every formula is $\tilde{\Pi}_{k+2}^b$ or lower.

Then by the constructions above there is a polynomial time PK_k^0 refutation which is well-formed provably in PV and where the initial cedents R' consist of $(\forall y\!<\!a\, \neg\phi(a, y))^\circ$ together with a table A of auxiliary clauses.

If $k \geq 1$ we add a $\bigwedge$-introduction step at the beginning to replace the initial cedent $(\forall y\!<\!a\, \neg\phi(a, y))^\circ$ in the refutation with the table $(\langle\neg\phi(a, r)\rangle)_{r<a}$. Let this new refutation be (Q, R, S, T, f, b) (we may assume that A is unchanged and that $|b|$ is polynomial in $|a|$). If $k = 0$ then, by definition, the translation $(\forall y\!<\!a\, \neg\phi(a, y))^\circ$ already has this form.

Let F be a polynomial time function listing all the clauses of the CNF we are refuting. That is, for each $r < a$, $F(r)$ outputs $\langle\neg\phi(a, r)\rangle$ as the complete

rth clause; if $r \geq a$, then $F(r)$ lists the contents of the $(r - a + 1)$st auxiliary clause. Let α be the polynomial time assignment that maps each literal to its truth value.

Then any witness to the instance $(F, \alpha, Q, R, S, T, f, b)$ of $1{-}\mathrm{Ref}(\mathrm{PK}^0_k)$ must be a witness to part 4 of the definition of $1{-}\mathrm{Ref}(\mathrm{PK}^0_k)$, namely it must be some clause $F(r)$ which is not satisfied by the assignment α. All the auxiliary clauses are true in α, so we must have $r < a$. Hence $\langle \neg\phi(a, r)\rangle$ must be false and r is a witness to our instance of $\forall\hat{\Sigma}^b_1(T^{k+2}_2)$. $\square$

5 A lower bound for $\overline{\mathrm{GI}}_3$

We translate the negation of $\mathrm{GI}_3(a)$ into propositional form. For simplicity we will assume that $a = 2^n$. Below, i ranges over $[1, a]$ and x, y, z, x', y', z' range over $[0, a)$. Our propositional variables are:

- $(U(x))_1, \ldots, (U(x))_n$ for all x, representing the bit graph of a function $U : a \to a$, that is, the interpretation of the variable $(U(x))_j$ is "the jth bit of the value of $U(x)$";
- $(V(y))_1, \ldots, (V(y))_n$ for all y, similarly for a function V;
- $(R_i(x))_1, \ldots, (R_i(x))_n$ for all $i < a$ and all x, for a family of functions R_i;
- $(S_{ix}(y'))_1, \ldots, (S_{ix}(y'))_n$ for all $i < a$ and all x and y', for a family of functions S_{ix};
- $(T_{ixy'}(z))_1, \ldots, (T_{ixy'}(z))_n$ for all $i < a$ and all x, y' and z, for a family of functions $T_{ixy'}$;
- $G_{ixy}(z)$ for all i, x, y and z, for a family of unary relations G_{ixy} on $[0, a)$.

We will write, for example, $U(x) = y$ as shorthand for the (small) conjunction $\bigwedge_{j\in[1,n]}(U(x))_j = (y)_j$ where each $(y)_j$ is 0 or 1 depending on the jth bit of y. Our propositional formula $\overline{\mathrm{GI}}_3$ is then the CNF:

1. $U(x) = y \to G_{1xy}(z)$ for all x, y, z;
2. $V(y) = z \to \neg G_{a0y}(z)$ for all y, z;
3. $R_i(x) = x' \wedge S_{ix}(y') = y \wedge T_{ixy'}(z) = z' \wedge G_{ix'y'}(z') \to G_{(i+1)xy}(z)$ for all i, x, y, z, x', y', z'.

We interpret $G_{ixy}(z)$ as expressing that the second player B wins the game G_i played with the moves x, y, z. So the principle states that: U is a winning strategy for B in G_1; playing 0 as the first move, and then using V, is a winning strategy for A in G_a; and the function families R_i, S_i and T_i together give a reduction of G_{i+1} to G_i for all i. In pictures:

$$\begin{tikzcd}
G_1: & x \arrow[r] & U \arrow[r] & y & & z \\
 & \vdots & & \vdots & & \vdots \\
G_i: & x' & & y' \arrow[d, "S_{ix}"'] & & z' \\
G_{i+1}: & x \arrow[u, "R_i"] & & y & & z \arrow[u, "T_{ixy'}"] \\
 & \vdots & & \vdots & & \vdots \\
G_a: & 0 & & y \arrow[r] & V \arrow[r] & z
\end{tikzcd}$$

The requirement that A's first move in the strategy for G_a is 0 makes the presentation simpler and makes no difference to the strength of the principle.

An alternative translation would be to use propositional variables to represent the graphs of our functions, rather than their bit graphs. We use bit graphs because we want our CNF to be narrow. Theorem 8 also holds for the other translation and the lower bound argument still works, if we change the definition of "constraining" below (define, for example, $R_i(x)$ to be constrained by a term if either the term includes "$R_i(x) = x'$" for some x' or includes "$R_i(x) \neq x'$" for at least $a/10$ many x's).

Lemma 22 $\overline{\mathrm{GI}}_3$ *has polynomial size* PK_1 *refutations.*

Proof Write H_{ixy} for $\bigwedge_z G_{ixy}(z)$ and $\neg H_{ixy}$ for $\bigvee_z \neg G_{ixy}(z)$.

From 1, by $\bigwedge$-introduction and then weakening we can derive $U(x') = y \to \bigvee_{y'} H_{ix'y'}$, for each x', y. Then, using for each x' a (polynomial size) depth n complete binary tree of resolutions on the bits of $U(x')$, we can derive

1*. $\bigvee_{y'} H_{1x'y'}$, for each x'.

Similarly from 2 by weakening and resolution we can derive

2*. $\neg H_{a0y}$, for each y.

From 3 we first derive $R_i(x) = x' \wedge S_{ix}(y') = y \to \neg H_{ix'y'} \vee G_{(i+1)xy}(z)$ for each i, x, y, z, x', y', and then from this get

3*. $R_i(x) = x' \to \neg H_{ix'y'} \vee \bigvee_y H_{(i+1)xy}$ for each i, x, x', y'.

Now for each x, we can derive $\bigvee_y H_{2xy}$ as follows: first, for each x' cut all instances of 3* with 1* to get $R_1(x) = x' \to \bigvee_y H_{2xy}$, then resolve to get rid of the assumptions $R_1(x) = x'$. By repeating this step we can eventually derive $\bigvee_y H_{ixy}$ for every x and i, contradicting 2*. $\square$

We now prove our resolution lower bound. We take PK_0^0 as the definition of the resolution system. The only rules are resolution and weakening. For some large a, suppose that Π is a resolution refutation of $\overline{\mathrm{GI}}_3(a)$. We will think of Π as a winning 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). The Prover wins when his term directly contradicts one of the clauses of $\overline{\mathrm{GI}}_3(a)$. 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 ρ on partial assignments to the variables of $\overline{\mathrm{GI}}_3$. We then show that if Π is small then with high probability a partial assignment ρ has several nice properties. We will choose one fixed ρ with these properties – in particular, if the Adversary's replies are consistent with ρ then the terms known by the Prover are always "thin" in a certain sense. This thinness allows the Adversary always to give answers which are consistent with $\overline{\mathrm{GI}}_3$, and hence the Prover is not able to win the game using the strategy Π.

In our random partial assignment ρ, and also the partial assignments α, α_0 and α' below (but not γ or γ', which represent the partial information known by the Prover), for each i and x either all bits of $R_i(x)$ will be given values, in which case we say that $R_i(x)$ is *set* to x' (where x' is the number with these bits) or none of the bits of $R_i(x)$ will be given values. Similarly for S, T, U and V. Hence we may treat all of these as partially defined (parametrized) functions. Also for each i, x and y, either G_{ixy} is set and has been given values for all z, or is not set and has no value for any z.

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

Our distribution is determined by two parameters, a probability p and a width w. We take p to be $a^{-9/10}$ and w to be $a/10$. We do not try to optimize the numbers. We define a random partial assignment ρ as follows.

1. For each pair $(i+1, x)$ with $i < a$, with probability p "select" $(i+1, x)$. Then for each "row" $i+1$ with $i < a$, suppose that distinct pairs $(i+1, x_1), \ldots, (i+1, x_m)$ were selected on that row. Randomly choose distinct numbers $x'_1, \ldots, x'_m$ (on row i) and use these to partially define R_i as a partial injection with domain the xs and range the x's. That is, for each j assign the bits of $R_i(x_j)$ with the bits of x'_j.

2. Similarly select each triple (i, x, y') for $i < a$ with probability p and then randomly define S_{ix} to be a partial injection from the selected y's on row i to a set of ys on row $i+1$.

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

4. Select each triple (i, x, y) (possibly with $i = a$) with probability p and then for each selected triple make each bit $G_{ixy}(z)$ true or false uniformly at random, with probability 1/2.

Proposition 23 (Chernoff bound) *Let X be the number of successes in a many independent random trials, each of which succeeds with probability q. Then for any $0 < \delta < 1$, $\Pr[X > (1+\delta)qa] < e^{-qa\delta^2/3}$.* $\square$

Lemma 24 *By the Chernoff bound, with exponentially high probability, for every i, $R_i(x)$ is set for at most $2pa$ values of x. Similarly for each function S_{ix} and $T_{ixy'}$. We may also assume that for each i and x, G_{ixy} is set for at most $2pa$ values of y, and that if G_{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 partial assignment ρ to set all variables U and V in a way which satisfies axioms 1 and 2 of $\overline{\mathrm{GI}}_3$.

5. For each x choose y at random amongst the $\geq a - 2pa$ values of y for which G_{1xy} was not set. Set $U(x) = y$ and set G_{1xy} to be true for all z.
6. For each y be on row a, there are two possibilities:

   Case 1: G_{a0y} is set. Then choose any z such that $\neg G_{a0y}(z)$. Set $V(y) = z$.

   Case 2: G_{a0y} is not set. Then choose z at random, set $V(y) = z$, make $G_{a0y}(z)$ false and make G_{a0y} true everywhere else.

Lemma 25 *With exponentially high probability, for any x and any $1 < i < a$, for all but $< 2a^2p^2$ numbers x' the following sets are disjoint in ρ: the domain of S_{ix}; the set of y' for which $G_{ix'y'}$ is set.*

Proof For any 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 each fixed x', there is any y' in an intersection is $\leq ap^2$. 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. $\square$

Lemma 26 *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 S_{1x}; the set of y' for which $G_{1x'y'}$ is set. Note that this second set includes $U(x')$.*

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

Definition 27 *We say a term t* constrains *$R_i(x)$ if the term contains any variable $(R_i(x))_j$. We define constraining $S_{ix}(y')$, $T_{ixy'}(z)$ and G_{ixy} similarly.*

*A term t is R-*fat *if $|\{(i,x) : t \text{ constrains } R_i(x)\}| > w$. S-fat, T-fat and G-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 28 *Any R-fat term t is falsified by ρ with exponentially high probability. Similarly for S, T and G.*

Proof A complication is that in each row the values ρ gives to each $R_i(x)$ are not set independently, since we insisted in the assignment that they should form a partial permutation. However if a literal $(R_i(x))_j$ (or its negation) is in t, it is falsified by ρ with probability at least $p/4$ since by Lemma 24 there were no more than $2pa$ values forbidden for $R_i(x)$, if it was set. Hence the probability that no such literal in t is falsified is $< (1-p/4)^w \leq e^{-pw/4} = e^{-a^{1/10}/40}$. $\square$

Definition 29 *An R-*path *in a partial assignment α is a maximal sequence $(j, x_j), (j-1, x_{j-1}), \ldots, (k, x_k)$ with $j > k$ such that $R_{j-1}(x_j) = x_{j-1}, \ldots, R_k(x_{k+1}) = x_k$.*

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

We say that (j, x_j) is at the bottom of the R-path and (k, x_k) is at the top, and the length of the path is $j-k$. The domain *of the path is $[k, j]$. Similarly for S-paths. An S-path τ is* full *if for each $i \in [l, m]$, $G_{ix_iy_i}$ is set, and for each $i \in [l, m-1]$, $T_{ix_{i+1}y_i}$ is total, and these hold in a way consistent with $\overline{\mathrm{GI}}_3$.*

Definition 30 *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, R and S are partial permutations for every choice of parameters.*
2. *$\rho \subseteq \alpha$.*
3. *$\gamma \subseteq \alpha$.*
4. *Every $R_i(x)$ constrained in γ is set in α.*
5. *For every $S_{ix}(y')$ constrained in γ, $R_i(x)$ and $S_{ix}(y')$ are set in α.*
6. *Every $T_{ixy'}(z)$ constrained in γ is set in α.*
7. *Every G_{ixy} constrained in γ is set in α.*
8. *Every S-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 G_{ixy} is true for all z, then $i \leq 2w+3$ and the R-path containing (i, x), if there is one, has all of its domain $\leq 2w+3$. We call this* uselessness.
11. *For no R-path in α are there more than two triples (i, x, x') such that (i, x') and $(i+1, x)$ are in the path and $R_i(x) = x'$ was set in ρ (rather than coming from γ). We call this* avoiding paths from ρ.

12. α is consistent with $\overline{\mathrm{GI}}_3$.

Lemma 31 *With polynomially high probability there is a completion α_0 of the empty assignment.*

Proof We need to show that we can extend ρ to make it full. The only obstacle to this is if there is some R-path on which the Gs and Ts were badly defined.

We may assume that there is no S-path in ρ of length two or more, by the following calculation. For any $i, x, x', x'', y, y', y''$, the probability that $(i+1, x)$, (i, x') and $(i-1, x'')$ form an R-path and $(i-1, x'', y'')$, (i, x', y') and $(i+1, x, y)$ form a matching S-path 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 S-path of length one such that G 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 $1 < i < a - 1$ the probability that these give an R-path and a matching S-path (on domain $[i, i+1]$) is $(p/a)^2$, and the probability that $G_{(i+1)xy}$ and $G_{ix'y'}$ 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 $i = a - 1$, if $x \neq 0$ then the calculation is as above. If $x = 0$, then G_{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 $(a, 0), (a-1, x')$ is a R-path, $(a-1, x', y'), (a, 0, y)$ is a matching S-path and $G_{(a-1)x'y'}$ is set, which happens with probability $< a^{3/10}/a^2$.

For $i = 1$, the probability that $G_{ix'y'}$ is set is slightly higher than p, since $G_{1x'U(x')}$ 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 may contain several S-paths of length 1, for which G may have been set by ρ at one end or the other, but not at both ends. Suppose $(i, x', y'), (i + 1, x, y)$ is such an S-path. $T_{ixy'}$ is partially defined, with domain of size $< 2pa$. If $G_{(i+1)xy}$ was set by ρ, then we can set $G_{ix'y'}$ to be false everywhere and extend $T_{ixy'}$ arbitrarily to a total function; this guarantees that (a) axiom 3 of $\overline{\mathrm{GI}}_3$ is satisfied and (b) $G_{ix'y'}(z')$ is false for some z', so that we do not violate uselessness. If $G_{ix'y'}$ was set by ρ and $i > 1$, then it is false for at least one z' so we can make $G_{(i+1)xy}$ false for one z and true everywhere else, set $T_{ixy'}(z) = z'$ and otherwise extend $T_{ixy'}$ arbitrarily; if $i = 1$ then possibly $G_{ix'y'}$ is true everywhere so we must make $G_{(i+1)xy}$ true everywhere, but in this case uselessness is not a problem. If neither is set by ρ then we make $G_{ix'y'}$ and $G_{(i+1)xy}$ both false everywhere.

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

Theorem 32 *For some $\varepsilon > 0$, there is no resolution refutation of $\overline{\mathrm{GI}}_3$ of size $< 2^{a^{\varepsilon}}$.*

Proof Suppose that there is a subexponential size proof Π. Then we can treat Π as a strategy for a Prover, as outlined above, and by Lemma 28 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 and add it to γ. We have shown that, provided that the Adversary's replies are consistent with ρ, then the Prover's strategy remains "thin" and γ constrains at most w many Rs, Ss, Ts and Gs.

The Adversary's strategy is to maintain a completion α of γ. As long as he does this, γ must be consistent with $\overline{\mathrm{GI}}_3$ 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'.

Suppose the Prover forgets a variable, so $\gamma' \subseteq \gamma$. Then the adversary can easily find a completion $\alpha' \subseteq \alpha$.

Suppose the Prover asks "what is $(R_i(x))_j$?". If $R_i(x)$ is set in α, then the Adversary answers appropriately and α is still a completion. Otherwise the Adversary choses a value x' for $R_i(x)$, uses that for his reply and adds it to α. We show that a suitable x' exists in Lemma 33 below.

Suppose the Prover asks "what is $(S_{ix}(y'))_j$?". Because of part 5 of Definition 30, we need to keep track of whether $S_{ix}(y')$ is constrained in γ even if $S_{ix}(y')$ is already set in ρ. So if $S_{ix}(y')$ is already constrained in γ, then it must be set in α and the Adversary answers appropriately. Otherwise first the Adversary uses Lemma 33 to set $R_i(x)$ (if it was not set already) and then uses Lemma 34 to set $S_{ix}(y')$ (if it was not set already).

If the Prover asks any bit of $T_{ixy'}(z)$ and it is not already set in α, the Adversary assigns values to all bits arbitrarily. If the Prover asks any bit of G_{ixy} and it is not already set in α, the Adversary assigns values to all bits arbitrarily, making at least one of them false to maintain uselessness. These steps could lead to a contradiction with $\overline{\mathrm{GI}}_3$ if the Ts or Gs lie on some S-path; but by fullness all the Ts and Gs on S-paths are already set by α in a consistent way. $\square$

Lemma 33 *There is a way to extend α to α' which sets a value for $R_i(x)$.*

Proof We divide the lemma into cases.

Case 1: Suppose $i > 1$. We list the properties which we do not want our choice x' of a value for $R_i(x)$ to satisfy, and give bounds on the number of x's with each bad property.

1. $R_i(\tilde{x}) = x'$ is in α for some $\tilde{x} \neq x$; $\leq 2pa + 2w$ ($2pa$ for Rs set by ρ, w for Rs constrained by γ and w for Ss constrained by γ – see part 5 of Definition 30).

2. $R_{i-1}(x')$ is set in α; $\leq 2pa + 2w$ (as above).

3. $G_{ix'y'}$ is set in γ for some y'; $\leq w$.

4. x' does not satisfy Lemma 25; $\leq 2a^2p^2$.

The sum of these bounds is less than a, so some good x' exists. We may extend α by setting $R_i(x) = x'$. Notice that by item 2, this does not grow any R-paths downwards, so uselessness is preserved; also by item 2 we continue to avoid paths from ρ.

We now need to add some more things to α to get fullness. By 1 and 2, (i, x') is not on any R-path in α. Hence any $G_{ix'y'}$ that is set was set either in ρ or in γ. By item 3, it can only have been set in ρ. By part 5 of Definition 30, since $R_i(x)$ has not been set yet in α, $S_{ix}(y')$ is not constrained in γ for any y', so $S^{\alpha}_{ix} = S^{\rho}_{ix}$. Hence by item 4, there is no y' such that $y' \in \text{dom}S^{\alpha}_{ix}$ and $G_{ix'y'}$ is set in α. So for every $(i, x', y'), (i+1, x, y)$ that is now (part of) an S-path in α matching our new (partial) R-path $(i+1, x), (i, x')$, we can set $G_{ix'y'}$ to be all false, set $G_{(i+1)xy}$ to be all false if it is not set already, and extend $T_{ixy'}$ arbitrarily to a total function, consistently with $\overline{\text{GI}}_3$ and with uselessness.

Case 2: Suppose $i = 1$. We list the properties which we do not want our choice x' of a value for $R_1(x)$ to satisfy, and give bounds on the number of x's with each bad property.

1. $R_1(\tilde{x}) = x'$ is in α for some $\tilde{x} \neq x$; $\leq 2pa + 2w$.
2. $G_{1x'y'}$ is set in γ for some y'; $\leq w$.
3. x' does not satisfy Lemma 26; $\leq 3a^2p^2$.

Fullness is preserved as in case 1.

For $y' = U(x')$, $G_{1x'y'}$ will be all true in ρ. But by the condition on avoiding paths from ρ and the limit on the number of Rs constrained by γ, the R-path containing $(1, x')$ has length at most $2w + 2$, so uselessness is preserved. $\square$

Lemma 34 *There is a way to extend α to α' which sets a value for $S_{ix}(y')$.*

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

Case 1: Suppose $i < a - 1$ and there is some x^* such that $R_{i+1}(x^*) = x$. We list the properties which we do not want our choice y of a value for $S_{ix}(y')$ to satisfy, and give bounds on the number of ys with each bad property.

1. $G_{(i+1)xy}$ is set in ρ or constrained by γ; $\leq 2pa + w$.
2. $S_{ix}(\tilde{y}) = y$ is in α for some $\tilde{y} \neq y'$; $\leq 2pa + w$.
3. $S_{(i+1)x^*}(y)$ is set in α; $\leq 2pa + w$.

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

By items 2 and 3, $(i+1, x, y)$ is not on any S-path in α, so together with item 1 this means that $G_{(i+1)xy}$ is not set in α. Hence there is no difficulty with extending $T_{ixy'}$, setting $G_{(i+1)xy}$ and possibly setting $G_{ix'y'}$ to achieve fullness. If $G_{ix'y'}$ is all true, then we must set $G_{(i+1)xy}$ to be all true. But this does not contradict uselessness, since the R-path containing (i, x') does not grow downwards.

Case 2: Suppose $i < a - 1$ and there is no x^* such that $R_{i+1}(x^*) = x$. The argument is as above, but simpler as we do not have to worry about $S_{(i+1)x^*}$.

Case 3: Suppose $i = a - 1$ and $x = 0$. Notice that by our construction of ρ for all y, G_{a0y} is set in ρ and $G_{a0y}(V(y))$ is false. Again we list the ys to avoid:

1. $S_{(a-1)0}(\tilde{y}) = y$ is in α for some $\tilde{y} \neq y'$; $\leq 2pa + w$.
2. G_{a0y} was set at random in ρ, at step 4 in the definition of ρ; $\leq 2pa$.
3. G_{a0y} was set in case 2 of step 6 in the definition of ρ, and $T_{(a-1)0y'}(V(y))$ is set in ρ; $\leq 2pa$ (by the Chernoff bound).
4. G_{a0y} was set in case 2 of step 6 in the definition of ρ, and $T_{(a-1)0y'}(V(y))$ is constrained in γ; $\leq w$.

Again some good y exists and we may extend α by setting $S_{(a-1)0}(y') = y$.

We may assume that $G_{(a-1)x'y'}$ is set and, by uselessness, that it is false for at least one z'. By item 2 and case 2 of step 6 of the definition of ρ, G_{a0y} is false for exactly one z, namely $z = V(y)$, and true everywhere else. By items 3 and 4, $T_{(a-1)0y'}(z)$ is not set for this z. Hence we may set $T_{(a-1)0y'}(z) = z'$ and then extend $T_{(a-1)0y'}$ arbitrarily to a total function. This gives fullness.

Case 4: Suppose $i = a - 1$ and $x \neq 0$. This is the same as case 2. □

Acknowledgments The authors would like to thank Arnold Beckmann, Emil Jeřábek, Jan Krajíček, Leszek Kołodziejczyk and Pavel Pudlák for helpful discussions and comments on earlier versions of this material.

References

[1] A. Atserias and M. Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, 2004.

[2] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998.

[3] A. Beckmann. Dynamic ordinal analysis. *Archive for Mathematical Logic*, 42(4):303–334, 2003.

[4] A. Beckmann and S. Buss. Polynomial local search in the polynomial hierarchy and witnessing in fragments of bounded arithmetic. Preprint, 2008.

[5] A. Beckmann and S. Buss. Characterizing definable search problems in bounded arithmetic via proof notations. Preprint, 2009.

[6] S. Buss. *Bounded Arithmetic.* Bibliopolis, 1986.

[7] S. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52(4):916–927, 1987.

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

[9] S. Buss. Chapter 1: An introduction to proof theory & Chapter 2: First-order proof theory of arithmetic. In S. Buss, editor, *Handbook of Proof Theory.* Elsevier, 1998.

[10] S. Buss and J. Krajíček. An application of Boolean complexity to separation problems in bounded arithmetic. *Proceedings of the London Mathematical Society*, 69:1–21, 1994.

[11] M. Chiari and J. Krajíček. Witnessing functions in bounded arithmetic and search problems. *Journal of Symbolic Logic*, 63(3):1095–1115, 1998.

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

[13] F. Ferreira. What are the $\forall\Sigma^b_1$-consequences of T^1_2 and T^2_2? *Annals of Pure and Applied Logic*, 75(1):79–88, 1995.

[14] J. Hanika. Herbrandizing search problems in bounded arithmetic. *Mathematical Logic Quarterly*, 50(6):577–586, 2004.

[15] E. Jeřábek. The strength of sharply bounded induction. *Mathematical Logic Quarterly*, 52(6):613–624, 2006.

[16] D. Johnson, C. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[17] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic*, 59(1):73–86, 1994.

[18] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Computational Complexity.* Cambridge University Press, 1995.

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

[20] J. Krajíček and P. Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 36(1):29–46, 1990.

[21] J. Krajíček, P. Pudlák, and G. Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.

[22] J. Krajíček, A. Skelley, and N. Thapen. NP search problems in low fragments of bounded arithmetic. *Journal of Symbolic Logic*, 72(2):649–672, 2007.

[23] J. Krajíček, P. Pudlák, and A. Woods. An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures and Algorithms*, 7(1):15–39, 1995.

[24] J. Krajíček and G. Takeuti. On induction-free provability. *Annals of Mathematics and Artificial Intelligence*, 6:107–126, 1992.

[25] A. Maciel, T. Pitassi, and A. Woods. A new proof of the weak pigeonhole principle. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 368–377, 2000.

[26] C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[27] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic*, number 1130 in Lecture Notes in Mathematics, pages 317–340. Springer, 1985.

[28] T. Pitassi, P. Beame, and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational complexity*, 3:97–220, 1993.

[29] P. Pudlák. A note on bounded arithmetic. *Fundamenta Mathematicae*, 136(2):86–89, 1990.

[30] P. Pudlák. Consistency and games - in search of new combinatorial principles. In V. Stoltenberg-Hansen and J. Väänänen, editors, *Logic Colloquium '03*, number 24 in Lecture Notes in Logic, pages 244–281. ASL, 2006.

[31] P. Pudlák. Fragments of bounded arithmetic and the lengths of proofs. *Journal of Symbolic Logic*, 73(4):1389–1406, 2008.

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

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

[34] A. Wilkie and J. Paris. On the scheme of induction for bounded arithmetic formulas. *Annals of Pure and Applied Logic*, 35:261–302, 1987.

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