Fragments of Bounded Arithmetic and the lengths of proofs

Pavel Pudlák *

December 12, 2007

Abstract

We consider the problem whether the $\forall\Sigma_1^b$ theorems of the fragments T_2^n form a strictly increasing hierarchy. We shall show a link to some results about the lengths of proofs in predicate logic that supports the conjecture that the hierarchy is strictly increasing.

1 Introduction

Bounded Arithmetic is, roughly speaking, Peano Arithmetic with induction restricted to formulas that define sets from the Polynomial Hierarchy. More precisely, we have also to include axioms that ensure that $x^{\lceil \log x \rceil}$ is a total function, in order to be able to talk about functions with values whose length is polynomial in the input length (in particular, polynomial time computable functions). This theory is called T_2 and its fragments corresponding to the levels Σ_n^p of the Polynomial Hierarchy are called T_2^n. It is an open problem whether the Polynomial Hierarchy is strictly increasing. What is only known is that it is strictly increasing with respect to an oracle. The state of art is the same in Bounded Arithmetic: it is an open problem if the hierarchy of theories T_2^n is strictly increasing, but a relativization is known with respect to which it is, [9]. The relativization of T_2^n is the theory $T_2^n[R]$ defined as follows. We augment the language of Bounded Arithmetic with a predicate symbol R and extend the axiom schema of induction in T_2^n to formulas of the same complexity containing the new predicate symbol R. We do not add any specific axioms about R.

Whereas the relativized separations of the levels is the best what we can prove about the Polynomial Hierarchy using current means, this is not the case with Bounded Arithmetic. We know that there are $\forall\Sigma_{n+2}^b[R]$ sentences that are provable in $T_2^{n+1}[R]$, but not in $T_2^n[R]$, however we do not know if the pair can be separated by sentences of smaller complexity. Although it is possible that the theories differ already in $\forall\Sigma_0^b[R]$ sentences, the smallest

*Institute of Mathematics, Academy of Sciences, Prague, Czech Republic. Supported by grant IAA1019401.

complexity for which it may be possible to prove separations with the current means is $\forall\Sigma_1^b[R]$. These sentences also have a very clear computational content—they define total NP-search problems.

In order to show such separations, some characterization of the $\forall\Sigma_1^b[R]$ theorems of $T_2^n[R]$ is needed. In [14] we have proposed combinatorial principles $\mathbf{A}_n$ based on finite games and proved that $\mathbf{A}_{n+1}$ is provable in T_2^n and every $\forall\Sigma_1^b$ sentence of T_2^n is provable from $S_2^1+\mathbf{A}_{n+5}$. In this paper we improve the above result by showing that already $S_2^1+\mathbf{A}_{n+1}$ proves all $\forall\Sigma_1^b$ sentences of T_2^n for $n=0,1,\ldots$. Thus we get a full characterization. These results also hold when relativized. It should be noted that quite recently Alan Skelley and Neil Thapen found simpler characterizations [15].

The main result of this paper concerns closely related principles that we denote by $\mathbf{A}_n^\infty$. These principles are inspired by the *adversary method* that is used to construct oracles. The adversary method is applied in the situations where we want to show that a problem cannot be solved by algorithms of a certain type. We consider an algorithm of the given type and put elements in the oracle so that the algorithm is unable to solve the problem. We can view this process as a two-player game in which one player is trying to solve the problem, but he does not know the oracle and has to ask about it the other player. The other player is trying to spoil it by picking the elements of the oracle in way that will prevent the first one to solve the problem.

The principle $\mathbf{A}_n^\infty$ is such a game for $\mathbf{A}_n$ with one essential modification: the numbers that the players use are not bounded. Due to this relaxation the number of queries that the first player has to make to solve the problem can be superexponential; it is a function that grows like n'-times iterated exponential, where n' grows linearly with n. This is proved using well-known bounds on cut-elimination in the predicate sequent calculus LK (or equivalently, on the size of Herbrand's disjunctions).

Thus principles $\mathbf{A}_n^\infty$ form a hierarchy that is increasing in the following sense. For every n, there exists $k>n$ such that $\mathbf{A}_k^\infty$ needs exponentially more queries than $\mathbf{A}_n^\infty$. We view it as evidence supporting the conjecture that sentences $\mathbf{A}_n[R]$ form a strictly increasing hierarchy, or equivalently that the sets of $\forall\Sigma_1^b[R]$ sentences of $T_2^n[R]$ increase with increasing n (put in yet another way that $T_2^{n+1}[R]$ is not $\forall\Sigma_1^b[R]$-conservative over $T_2^n[R]$). It is, however, still an open problem whether this is indeed true.

We shall assume that the reader is familiar with basic concepts and results of proof complexity as presented eg. in [7].

2 Herbrand's Theorem

We shall not use Herbrand's Theorem directly, but it is simpler to explain some basic ideas and notation on this more elementary case. We shall work in the predicate calculus without equality, but eventually, in Section 4, we shall apply our results to the Quantified Propositional Calculus.

The following is a well-known form of Herbrand's Theorem.

Theorem 1 *A sentence of the form $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \phi(x_1, y_1, x_2, y_2, \ldots)$ is logically valid iff*

$$\bigvee_i \phi(t_{i1}, y_{i1}, t_{i2}, y_{i2}, \ldots) \tag{1}$$

is a propositional tautology, where

1. *terms t_{ij} and variables y_{ij} can be linearly ordered so that*
   (a) $y_{i,j} < t_{i,j'}$ *(resp. $y_{i,j} > t_{i,j'}$) if $j < j'$ (resp. $j > j'$),*
   (b) *if $y_{ij} < t_{i'j'}$, then y_{ij} does not occur in $t_{i'j'}$;*
2. *if y_{ij} is the same variable as $y_{i'j'}$, then $j = j'$ and*

$$t_{i1} = t_{i'1}, y_{i,2} = y_{i'2}, \ldots, y_{ij} = y_{i'j}.$$

This rather complicated condition on terms is automatically satisfied in the more common form of Herbrand's Theorem, in which one uses new function symbols $f_1, f_2, \ldots$ and terms containing these symbols. Then the Herbrand disjunction has the form

$$\bigvee_i \phi(s_{i1}, f_1(s_{i1}), s_{i2}, f_2(s_{i1}, s_{i2}), \ldots)$$

and we do not have to state any additional conditions.

To obtain the above theorem from the classical form of Herbrand's Theorem, replace all maximal subterms that start with the new function symbols $f_1, f_2, \ldots$ by new variables (the same terms by the same variables). The ordering mentioned in the above form of the theorem is simply the ordering of terms by their size, where for terms of equal size we can choose an arbitrary order).

Notice also that if we start with the sequent

$$\longrightarrow \ldots, \phi(t_{i1}, y_{i1}, t_{i2}, y_{i2}, \ldots), \ldots \tag{2}$$

instead of the disjunction (1), we can introduce quantifiers in the order reverse to the ordering of terms and variables and thus we derive the sequent

$$\longrightarrow \forall x_1 \exists y_1 \forall x_2 \exists y_2 \ldots \phi(x_1, y_1, x_2, y_2, \ldots). \tag{3}$$

Turning this argument around, we can show that Theorem 1 follows from the Middle Sequent Theorem. Indeed, since the middle sequent (2) is derived only using propositional rules, it is a tautology. To get the ordering of terms and variables in Theorem 1, take the order in which quantifiers are introduced in the opposite direction.

To apply the above theorem (in fact a more complicated theorem below), we need a more transparent way of representing the dependences of terms on variables. We shall use tables in which rows correspond to the disjuncts and the ordering of columns from left to the right determines the ordering of terms and variables. A column may contain more terms in case their ordering is irrelevant.

Example 1. Suppose our formula is

$$\forall x_1 \exists y_1 \forall x_2 \phi(x_1, y_1, x_2).$$

Suppose the Herbrand disjunction has four terms. Then one possible dependence of terms on variables is given by

α_1	b_1							$\alpha_{51}(b_1, b_2, b_3, b_4)$
		$\alpha_2(b_1)$	b_2					$\alpha_{52}(b_1, b_2, b_3, b_4)$
				$\alpha_3(b_1, b_2)$	b_3			$\alpha_{53}(b_1, b_2, b_3, b_4)$
						$\alpha_4(b_1, b_2, b_3)$	b_4	$\alpha_{54}(b_1, b_2, b_3, b_4)$

Here b_i stand for variables, and α_j for terms.

The condition 1.(b) says that a term can depend only on variables that precede it in the ordering. In this notation we indicate that explicitly mentioned variables are those that *may* occur in the term, but in general they do not have to.

Example 1-cntd. Since the ordering determines the dependence, we can represent the above table more schematically by:

A	B							A
		A	B					A
				A	B			A
						A	B	A

Another concept that helps understanding complicated relations between terms is the concept of a game. A *game of length* k is a subset of a Cartesian product

$$G \subseteq M_1 \times M_2 \times \ldots \times M_k.$$

The set M_i is the *set of possible moves in the* i*-th step* of the game. The game is played by two players who alternate in choosing elements form sets M_i. G is interpreted as the set of winning positions of the first player. Every finite game has a winning strategy either for the first player or for the second. We shall call the two players, as usual, Alice and Bob (but who is the first and who is the second will depend on the specific game).

More generally, we can allow a player to play several moves before the other player plays. This can be formally represented by having the sets M_i to be sets of strings. We define the length of such a game to be the number of alternations, ie., k.

We can interpret a Herbrand disjunction as a strategy for playing a game. The terms form a strategy for the moves of Alice, the variables are moves of Bob. A winning position for Alice are the values that make the disjunction true. Thus if the disjunction is a propositional tautology, the terms define a *winning* strategy for Alice.

The game that we get from Herbrand's Theorem has an important inner structure. We can view it as composed from several copies of the same game. The copies correspond to the disjuncts. To avoid confusion, we shall extend our terminology. A game composed from several games will be called a *match*. In general a match can be composed from copies of several games. We shall say that the copies of games are played on several *boards*.

Formally, a match is a string $G_1, \dots, G_l$ of games of the same length k and an $n \times m$ matrix S, called a schedule, with the following properties: the rows of the matrix are labeled by games, the entries of S are A,B, or empty, and each row has k alternating occurrences of A and B.[1]

Example 1-cntd. The matrix of Example 1 is a schedule of a match composed from games of length 3 played on 4 boards. Which games are played on the boards was not specified, but if it represents a Herbrand disjunction, it is the same game on all boards.

Suppose a match M with games $G_1, \dots, G_l$, n boards and a schedule S is given. Suppose we keep games as they are assigned to boards, but change the order of moves of Alice and Bob. Let M' be the match with the changed schedule S'. Then we can ask, what is the relation between the two matches. In particular, we shall say that S' is *harder* (for Bob) than S, if the following holds true: if for every Alice's strategy for M, Bob can win on a subset of boards D, then also for every Alice's strategy for M', Bob can win on the subset of boards D.

The question which schedules are hard is interesting in connection with Herbrand's theorem (see a result concerning sentences with prefixes $\exists\forall\exists$ in [14]). We shall not consider that problem here, we only need the following simple result.

Lemma 1 *Suppose that in a schedule S we have A in a row i and a column j and B in a row i' and column $j+1$, where $i \neq i'$. Then, if we switch columns j and $j+1$, ie., move A to position $(i, j+1)$ and move B to position (i', j), we obtain harder schedule.*

Proof. Let S' denote the schedule obtained by switching the columns. Let σ be Alice's strategy for the match with schedule S. Then we σ is also a strategy for the match with schedule S'. Hence winning moves for Bob in the match with S' are also winning moves in the match with S. ∎

We shall also need to consider formulas with blocks of same quantifiers. To describe the dependences of terms in Herbrand proofs of such sentences we have to consider more general games and more general schedules. In such games a player can play a series of moves, then the other can play a series of moves and so on. Then one can define matches in which a series of moves of one player on a given board is interrupted by moves of the other player on other boards. Due to the previous lemma, we do not have to consider such schedules.

Corollary 2 *Every schedule S can be transformed into a harder schedule S' in which players play their moves on each board in blocks, ie., without being interrupted by the other player.*

[1] One can define a more general concept of a match by using games of different lengths, but we do not need such matches.

It is good to realize what it means in terms of proofs. Applying the corollary to Herbrand's theorem we obtain:

Corollary 3 *Suppose ϕ is a logically valid sentence with k blocks of alternating quantifiers. Then ϕ has a cut-free proof in which quantifiers are always introduced in blocks.*

3 An auxiliary result

In this section we shall prove a general statement that we shall later use for showing that principles $\mathbf{A}_k$ characterize $\forall\Sigma_1$ consequences of T_2^k. We shall prove it for LK, the classical sequent calculus. A similar result was recently proved by Stefan Hetzl [5].

Lemma 2 *Let d be an LK proof of an existential sentence*

$$\exists x_1 \dots \exists x_n \psi(x_1, \dots, x_n)$$

(i.e., ψ an open formula). Suppose that d satisfies the distinct eigenvariables condition. Then one can construct in polynomial time an open formula $\Theta(y_1, \dots, y_m)$, terms t_{ij} and a propositional proof d' of

$$\Theta(y_1, \dots, y_m) \to \bigvee_{i=1}^{p} \psi(t_{i1}, \dots, t_{in}),$$

such that

1. $\vdash \exists y_1 \dots \exists y_m \Theta(y_1, \dots, y_m)$,
2. *formula Θ has a special form defined below.*

We are assuming that the language contains a constant symbol (alternatively, we can allow parameters in Θ). We can, moreover, show that d' is cut-free, but we shall not need it. What we shall actually need is a condition stronger than having a polynomial algorithm, we need *the provability of the existence of d' in S_2^1.* The construction of d' and Θ consists of simple manipulations with terms. Such operations are well-known to be formalizable in S_2^1, thus the stronger property can be verified by inspecting the proof below.

The proof is based on a simple and well-known observation that one can avoid the use of cuts by introducing implications, or as we prefer, by introducing negations and disjunctions. Indeed, from sequents

$$\Gamma \longrightarrow \phi, \Delta \quad \text{and} \quad \phi, \Gamma' \longrightarrow \Delta'$$

we get

$$\phi \vee \neg\phi, \Gamma, \Gamma' \longrightarrow \Delta, \Delta'$$

using left $\neg$–introduction and left $\vee$–introduction, thus *without using cut.* In order to obtain quantifier free formulas we furthermore omit all quantifier rules. Then we have to omit also contractions that are applied to the the formulas obtained after quantification. Thus we get multiple copies of term instances of the matrices of cut formulas, which we collect into conjunctions, and multiple copies of instances of the matrix of ψ, which we collect into the disjunction. (A *term instance of a formula* ϕ is a formula obtained from ϕ by substituting terms (including terms that are mere variables) for free variables of ϕ.) A more complicated part of the lemma is the explicit description of Θ.

Description of Θ. It has the form

$$\bigwedge_{\ell}(\bigwedge_{j}\phi^0_{\ell j} \vee \bigwedge_{k}\neg\phi^1_{\ell k}),$$

where

1. for every ℓ, $\phi^0_{\ell j}$ and $\phi^1_{\ell k}$ are term instances of the same quantifier free formula ϕ_ℓ such that
   - $\phi^0_{\ell j}$ has the form $\phi_\ell(Y_1, S_2, Y_3, \ldots)$ and
   - $\phi^1_{\ell k}$ has the form $\phi_\ell(S_1, Y_2, S_3, \ldots)$,

   where Y_r are blocks of variables from $y_1, \ldots, y_m$ and S_r are blocks of terms, the lengths of blocks Y_r and S_r being the same;
2. the number of alternations of blocks is bounded by the highest quantifier complexity (i.e., the number of quantifier alternations) of formulas in proof d;
3. the variables $y_1, \ldots, y_m$ and the occurrences of terms (see below) in Θ are linearly ordered by a relation $<$ so that
   (a) the order $<$ respects the order of blocks, ie., if $\hat{s}$ is an occurrence of a term s in S_r, y_h occurs in $Y_{r'}$, and $r < r'$ (resp. $r > r'$), then $\hat{s} < y_h$ (resp. $\hat{s} > y_h$);
   (b) if $\hat{s} < y_h$, then y_h does not occur in term s,
   (c) every variable y_h may occur at most once in some block Y_r in $\phi_{\ell j}$; it may occur more than once in different blocks Y_r, but all occurrences must be in term instances of the same ϕ_ℓ, and the prefix of terms and variables before the occurrence of y_h must always be the same.

An occurrence of a term s is formally a pointer to a place in Θ where s occurs; we are using $\hat{s}$ to denote an occurrence of s.

For proving this lemma we shall need the following simple fact.

Lemma 3 *Every proof d in LK can be transformed in polynomial time into a proof d' of the same sequent such that the cut formulas of d' are in prenex form and the maximal quantifier complexity of formulas in d' is the same as in d.*

Proof. It is easy to extend the proof so that the initial sequents are of the form $\alpha \longrightarrow \alpha$ for α *quantifier free.*

Consider the cut formulas of d. For every such formula ϕ pick a prenex form that has the same quantifier complexity as ϕ. Then for every subformula ψ of a cut formula ϕ, we take the prenex form of ψ whose prefix of quantifiers is a part of the prefix of ϕ, i.e., the order of quantification of the bound variables of ψ is the same in both prenex formulas. We shall denote the chosen prenex form of ψ by $Pren(\psi)$. We choose the prenex form of a term instance of ψ to be the corresponding term instance of $Pren(\psi)$ and we extend the notation $Pren$ to these formulas.

To obtain d' from d, we first replace all term instances of subformulas of cut formulas by their chosen prenex forms. By doing so we only damage the rules that introduce propositional connectives. Therefore, for each connective, we shall add a small subproof that will fix it. We shall explain it only on the left $\neg$–introduction. Suppose

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

is a an instance of this rule in d. After replacing formulas by prenex forms, this becomes

$$\frac{\Gamma' \longrightarrow Pren(\phi), \Delta'}{\Gamma', Pren(\neg\phi) \longrightarrow \Delta'}.$$

To fix it, we first derive $Pren(\phi), Pren(\neg\phi) \longrightarrow$. This sequent has a cut-free proof of size polynomially bounded in the length of ϕ and it is easy to construct (we leave it to the reader). Then we apply cut instead of $\neg$-introduction:

$$\frac{\Gamma' \longrightarrow Pren(\phi), \Delta' \qquad Pren(\phi), Pren(\neg\phi) \longrightarrow}{\Gamma', Pren(\neg\phi) \longrightarrow \Delta'}.$$

Notice that the cut formula is in prenex form.

■

Proof. [of Lemma 2] Let a proof d be given. By the previous lemma we may assume that only prenex formulas are used in cuts. Further, we may assume that all cut formulas are closed, as we can substitute a constant for all free variables of the proof. Finally, since we use tree-like proofs, we may assume the *distinct eigenvariable condition:* each instance of a strong quantifier rule uses a different eigenvariable.

We shall describe how to transform d into d'. We shall process d starting from initial sequents to the end sequent.

1. When we reach an instance of a quantifier rule, we omit it.
2. Because of that some formulas that became the same after quantification remain different, hence we cannot perform all contractions as in the original proof. When we reach such an instance of contraction, we omit it and keep the distinct formulas in

the sequent as they are. Since the principal formulas of cuts are in prenex form, once a quantifier rule is applied to a formula α of d, no propositional rules are applied to α anymore. Hence the only rule that is affected by the presence of multiple term instances of the matrices of cut formulas are cuts.

3. Consider an instance of a cut rule with the cut formula Φ_ℓ in d:

$$\frac{\Gamma \longrightarrow \Phi_\ell, \Delta \qquad \Gamma', \Phi_\ell \longrightarrow \Delta'}{\Gamma, \Gamma' \longrightarrow \Delta, \Delta'}.$$

Let ϕ_ℓ denote the matrix of Φ_ℓ. Let $\phi^1_{\ell,k}$ be the term instances of ϕ_ℓ in the consequent of the left premise and $\phi^0_{\ell,j}$ be the term instances of ϕ_ℓ in the antecedent of the right premise. Hence in the new proof the premises have the form

$$\Gamma \longrightarrow \ldots \phi^1_{\ell,k} \ldots, \Delta \qquad \Gamma', \ldots \phi^0_{\ell,j} \ldots \longrightarrow \Delta'.$$

Applying a series of $\neg$-introductions to the left premise, a series of $\wedge$-introduction to both premises and, finally, one $\vee$-introduction we get

$$\Gamma, \Gamma', \bigwedge_j \phi^0_{\ell j} \vee \bigwedge_k \neg\phi^1_{\ell k} \longrightarrow \Delta, \Delta'.$$

Thus we obtain a valid proof d'. We now need to check the properties of the end sequent. The consequent consists of formulas $\psi(t_{i1}, \ldots, t_{in})$, the antecedent consists of formulas $\bigwedge_j \phi^0_{\ell j} \vee \bigwedge_k \neg\phi^1_{\ell k}$. The blocks of variables and terms in $\phi_\ell(Y_1, S_2, Y_3, \ldots)$ and $\phi_\ell(S_1, Y_2, S_3, \ldots)$ correspond to blocks of alternating quantifiers. Variables in blocks $Y_1, Y_2, \ldots$ are eigenvariables of strong quantifier inferences, terms in blocks $S_1, S_2, \ldots$ are terms in of the weak quantifier inferences.

We define an ordering $<$ on these variables and terms as follows. Consider the order in which quantifiers are introduced. Since the proof is a tree, this is a partial ordering. An introduction of a strong quantifier corresponds to a specific variable y_i, since we are assuming the distinct eigenvariable condition. An introduction of a weak quantifiers is associated with a term; the same term may occur in different context, thus we have to consider term occurrences. To get the ordering $<$ needed in the lemma, we take the reverse ordering and extend it to a linear ordering in an arbitrary way.

Conditions 3. (a), (b), (c) are the basic properties that have to be satisfied when quantifiers are introduced. Namely, (a) is the fact that quantifiers of a quantifier prefix of a formula must be introduced in the reverse order; (b) corresponds to the condition that the eigenvariable must occur only on the position where it is going to be quantified; similarly (c) expresses the fact that we have to contract all formulas with an eigenvariable, before we apply a strong rule.

It remains to prove that $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$ is logically valid. We shall prove it after we give an interpretation of Θ in terms of games; it will be clearer then. ∎

Interpreting Θ in terms of games.

Since we have interpreted Herbrand's disjunctions, the interpretation of the lemma is easy. We interpret Θ as a match played by two players Alice and Bob, with a given strategy for Alice. The match has the following form and rules.

1. The match uses some games $G_1, \ldots, G_q$.
2. Each game G_ℓ has to occur at least once in the form in which Alice starts and at least once in the form in which Bob starts.
3. Bob has to follow Rule $\mathcal{B}$ (to be defined below).
4. Alice wins the match if at least for one ℓ she succeeds to win on at least one board with G_ℓ where she starts and on one board with G_ℓ where Bob starts. (Otherwise wins Bob.)

Rule $\mathcal{B}$ is the following.

> *If Bob is to move and the situation on the current board is the same as it is on another board where the same game is played, then he has to play in the same way on both (all such) boards.*

Perhaps more intuitively: *for every ℓ, Bob has to apply the same strategy of playing G_ℓ on all boards on which G_ℓ is played.*

This is the description of the match. Now we shall explain the relation to Θ. To this end suppose that a structure M for the language used in Θ is given.

- For every formula ϕ_ℓ we have one game G_ℓ, which is defined by ϕ_ℓ. Moves on this game are strings of elements of M of lengths equal to the corresponding blocks of variables. Thus the length of the game G_ℓ is the number of quantifier alternations in Φ_ℓ.
- The two dual occurrences of G_ℓ correspond to the presence of a term instance of ϕ_ℓ and $\neg\phi_\ell$.
- The schedule of the match is given by the ordering of variables and terms. Terms correspond to Alice's moves, variables to Bob's moves. Property 3. (a) of Θ guarantees that games are played in the correct order.
- Property 3. (b) ensures that terms of Θ define a strategy of Alice, (namely, that the next Alice's move defined by terms depends only on moves played before).
- The fact that a variable y_i occurs several times means that Bob has to play in the same way in those positions. According to property 3. (c), this can happen only in certain situations. Rule $\mathcal{B}$ expresses this condition in terms of games.
- $\Theta(u_1, \ldots, u_m)$ is true for some $u_1, \ldots, u_m \in M$, iff the moves determined by these elements beat Alice's strategy determined by the terms.

Now we can finish the proof of Lemma 2. We need to show that $\exists y_1, \ldots, \exists y_m\, \Theta(y_1, \ldots, y_m)$ is true in every model. Let a model M be fixed. Interpret Θ in terms of games and matches as described above. Since the games $G_1, \ldots, G_q$ have finite numbers of rounds, they are determined. We define a strategy for Bob to win the match. For every ℓ, if there is a winning strategy for the first player in G_ℓ, Bob shall pick such a strategy and play it on all boards where this game is played and where he starts the game. On the boards with G_ℓ where Alice starts, he will play an arbitrary strategy, say, he will play the same element all the time. If there is a winning strategy for the second player in G_ℓ, he will play it on all boards where this game is played and where Alice starts, and he will play an arbitrary strategy on the boards where he starts. Thus he will win the match. The elements $b_1, \ldots, b_m$ that he will play will satisfy Θ. ■

Let us note that it is also possible to give a direct proof, which we shall do in Section 5. The lengths of proofs of formulas Θ will play an important role.

4 The principle $\mathbf{A}_k$.

Now we turn to the main topic of the paper, which are principles that characterize $\forall\Sigma^b_1$ sentences of the fragments T^k_2 of Bounded Arithmetic. The principles will be based on the matches described in the previous section. We need, however, that the games and strategies are computable in polynomial time.

Thus we consider games $G \subseteq M_1 \times M_2 \times \ldots \times M_k$, $M_j = \{0,1\}^{n_j}$, which are defined by boolean circuits $C(\bar{x}_1, \ldots, \bar{x}_k)$ (ie., $G = \{(\bar{x}_1, \ldots, \bar{x}_k);\ C(\bar{x}_1, \ldots, \bar{x}_k) = 1\}$). Strategies are also defined by boolean circuits. We shall say that a schedule is a *schedule for an $\mathbf{A}_k$-match,* if it has the form described in the previous subsection and the games have lengths at most k. We shall also require the Rule $\mathcal{B}$.

Definition 1 (Principle $\mathbf{A}_k$.) *Given a schedule for an A_k-match, given circuits for the games and for a parallel strategy of Alice, there exist boolean vectors that represent winning moves of Bob (ie., such that for every game G_ℓ, Bob wins on all boards with G_ℓ where he starts or on all boards with G_ℓ where Alice starts).*

Principle $\mathbf{A}_k$ was introduced in [14]. In that paper we discussed it at length, gave some examples and also considered a special version of these principles. Therefore we shall proceed immediately to our characterization of the $\forall\Sigma^b_1$ sentences of T^k_2. The first part was proved already in [14].

Proposition 4 *For all $k \geq 1$, $\mathbf{A}_{k+1}$ is provable in T^k_2, and $\mathbf{A}_1$ is provable in S^1_2.*

Theorem 6.1 of [14] states that for all $k \geq 1$, $\mathbf{A}_k$ is provable in $S^k_2(PV)$. The above proposition is an immediate corollary, because theories $S^{k+1}_2(PV)$ are Σ^b_1-conservative over T^k_2, and $S^1_2(PV)$ is Σ^b_1-conservative over S^1_2. In the other direction we only proved that all $\forall\Sigma^b_1$ sentences of T^k_2 are provable in $S^1_2 + \mathbf{A}_{k+4}$. Our new result, with the proposition above, gives an exact characterization.

Theorem 5 *For all $k \geq 1$, all $\forall\Sigma_1^b$ sentences of T_2^k are provable in $S_2^1 + \mathbf{A}_{k+1}$.*

Our proof is based on a different idea than the proof of the weaker result of [14]. We shall use a characterization of $\forall\Sigma_1^b$ sentences of T_2^k by certain reflection principles in quantified propositional calculus proved in [8]. Let us briefly recall these concepts.

Quantified propositional logic is an extension of propositional logic based on formulas in which propositional variables can be quantified. The proof system G for quantified propositional logic is essentially the sequent calculus LK that uses the language of quantified propositional logic. The only difference between LK and G is that in the quantified propositional calculus G the role of terms is played by quantifier-free propositional formulas. Thus, for example, the rule of the right $\exists$-introduction is

$$\frac{\Gamma \longrightarrow A(B), \Delta}{\Gamma \longrightarrow \exists x\ A(x), \Delta},$$

where B is a proposition without quantifiers. The classes Σ_k^q of quantified propositional formulas is defined by restricting the quantifier alternations to k with existential quantifiers coming first. The subsystems G_k of G are defined by restricting G to Σ_k^q formulas. In G_k^* we furthermore require the proofs to be in the tree-like form. (Here we are studying proof complexity, so such restrictions play important roles.)

The reflection principle $\Sigma_1^q RFN(G_k^*)$ says that G_k^* proves only true Σ_1^q formulas. It can be formalized as a $\forall\Sigma_1^b$ sentence of bounded arithmetic. The following proposition has never been stated explicitly, but it easily follows from well-known results.

Proposition 6 *For all $k \geq 1$, all $\forall\Sigma_1^b$ sentences of T_2^k are provable in $S_2^1 + \Sigma_1^q RFN(G_{k+1}^*)$.*

Proof. In [8] (Corollary 7.1) we proved that for $k \geq j \geq 2$, $\forall\Sigma_j^b$ sentences of T_2^k are axiomatized by $S_2^1 + \Sigma_j^q RFN(G_k)$. The proof also gives that $S_2^1 + \Sigma_j^q RFN(G_k)$ proves all $\forall\Sigma_j^b$ sentences of T_2^k for all $k \geq j \geq 1$. In order to obtain the proposition, we have to replace G_k by G_{k+1}^*. To this end, it suffices to replace the fact that G_k simulates T_2^k (in the sense of [8]) by the result that G_{k+1}^* simulates S_2^{k+1}, hence it also simulates T_2^k, see [10] and [7] (Theorem 9.2.6).

■

Hence to prove Theorem 5, it suffices to prove the following lemma.

Lemma 4 *For all $k \geq 2$, $S_2^1 + \mathbf{A}_k \vdash \Sigma_1^q RFN(G_k^*)$.*

Proof. We shall show that $\Sigma_1^q RFN(G_k^*)$ follows from $\mathbf{A}_k$. The following argument can be formalized in S_2^1.

Let d be a G_k^* proof of a Σ_1^q sentence $\Psi = \exists x_1 \dots \exists x_n \psi(x_1, \dots, x_n)$. By Lemma 2 adapted to the propositional sequent calculus, we can construct a propositional proof d' of the proposition

$$\Theta(y_1, \dots, y_m) \rightarrow \bigvee_{i=1}^{p} \psi(t_{i1}, \dots, t_{in}). \tag{4}$$

When we interpret Θ in terms of games, it defines a schedule of an $\mathbf{A}_k$-match. The games are defined by boolean formulas (which are a special case of circuits), and so is Alice's strategy. Thus Θ defines an instance of the principle $\mathbf{A}_k$.

If we assume the truth of $\mathbf{A}_k$, we can find an assignment $\bar{b}$ to the boolean variables $y_1, \ldots, y_m$ such that $\Theta(\bar{b})$ is true. Now we substitute $\bar{b}$ into (4) and interpret 0's and 1's as constants for truth and falsehood. Thus we have a proof in the propositional sequent calculus of

$$\Theta(\bar{b}) \rightarrow \bigvee_{i=1}^{p} \psi(t_{i1}[\bar{y}/\bar{b}], \ldots, t_{in}[\bar{y}/\bar{b}]). \tag{5}$$

Since S_2^1 proves the reflection principle for propositional sequent proofs, proposition (5) is true. Since $\Theta(\bar{b})$ is true, also the consequent of (5) is true. Whence Ψ is true.

Thus, assuming $\mathbf{A}_k$, we have proved that if a Σ_1^q sentence has a proof in G_k^*, then it is true. This is precisely the $\Sigma_1^q RFN(G_k^*)$. ∎

5 Playing against an oracle

The main problem about $\forall\Sigma_1^b$ sentences provable in the fragments T_2^k is whether T_2^{k+1} proves the same $\forall\Sigma_1^b$ as T_2^k, ie., whether the former theory is $\forall\Sigma_1^b$-conservative over the latter. We conjecture that the answer is negative, but proving that seems hopelessly difficult. So our aim is to show at least some indirect that confirms this conjecture.

Let us recall that the problem about the $\forall\Sigma_1^b$-conservativity can be reduced to a purely computational problem. Let $\forall x \exists y\ \phi(x, y)$ be a true $\forall\Sigma_1^b$ sentence. We associate with this sentence a *total polynomial search problem:* for a given x, find y such that $\phi(x, y)$ is true. Given such a sentence Φ, we shall denote by $S(\Phi)$ the associated search problem. It is a well-known fact that if $S_2^1 \vdash \Phi \rightarrow \Psi$, where Φ and Ψ are $\forall\Sigma_1^b$, then $S(\Psi)$ can be reduced to $S(\Phi)$, see [4]. The following lemma states this fact more precisely.

Lemma 5 *If $S_2^1 \vdash \forall x \exists y \phi(x, y) \rightarrow \forall x \exists y \psi(x, y)$, then for every given x it is possible to find a witness y satisfying $\psi(x, y)$ using an interactive algorithm that runs in polynomial time and uses witnesses y_i satisfying $\phi(x_i, y_i)$, for x_i computed in the computation. (Since the computation is polynomial, the number of queries x_i is also polynomial.)*

Since we know that the sentences $\mathbf{A}_{k+1}$ characterize $\forall\Sigma_1^b$ sentences of T_2^k, in order to prove that T_2^{k+1} is not conservative over T_2^k, it suffices to prove that $S(\mathbf{A}_{k+1})$ is not reducible (in the sense of the lemma) to $S(\mathbf{A}_k)$. We conjecture that it is so, but proving that is beyond current means. Notice that if $P = NP$, then all polynomial search problems (even non-total) are solvable in polynomial time, hence they are trivially reducible to each other.

What seems a more feasible goal is to show that $S(\mathbf{A}_{k+1})$ is not reducible to $S(\mathbf{A}_k)$, for $k = 1, 2, \ldots$, when this problem is relativized to an oracle. That would imply our conjecture for the *relativized fragments,* namely, it would imply that $T_2^{k+1}[R]$ is not $\forall\Sigma_1^b[R]$-conservative over $T_2^k[R]$. The construction of such oracles still an open problem, though some progress has been achieved; in particular such oracles are known for $S^1_2[R]$ versus $T^1_2[R]$, cf. [1], and for $T^1_2[R]$ versus $T^2_2[R]$, cf. [6, 12].

In this section we shall show a different type of evidence that the principles $\mathbf{A}_k$ form an increasing hierarchy. The idea is to consider $\mathbf{A}_k$-matches that are not defined explicitly, but only by answers of an oracle. Furthermore, the moves are not restricted to a finite domain, one can play an arbitrary natural number. We shall study the minimal number of oracle queries that are needed to find moves that beat Alice's strategy. We shall show that this number increases as several times iterated exponential function. The number of iterations of the exponential function increases linearly with k.

Now we shall describe this game, which we denote by $\mathbf{A}^\infty_k$, in more detail. An instance of the game is given by an $\mathbf{A}_k$-schedule. There are two players which we call *We* and *Oracle.* We can ask queries of the form

1. *what would Alice play in a particular position in the match?*

2. and *who is winning in a particular final position on a particular board?*

Oracle may answer the first type of question by an arbitrary number and it can choose Alice or Bob arbitrarily in case of the second question. The Oracle pretends that it uses one fixed strategy of Alice and that each game of the match is are fixed. In other words, once a query is answered, then that particular part of Alice's strategy or that particular final position of a game is fixed for the rest of the communication.

Our goal is to find a winning position of Bob in the match played according to the imaginary Alice's strategy. What it amounts to is to ask a lot of queries about various plays of the match so that We acquire information about games and the strategy that is sufficient for constructing winning moves of Bob. The aim of Oracle is to prevent Us to do that.

We shall first note that We can always win, in spite of all the advantage that Oracle has. This can be proved by a simple non-constructive argument, see [14], which does not give any explicit bounds on the number of queries. Since the number of queries is of concern to us, we shall reinterpret the match in terms of the formula Θ and use bounds on the lengths of proofs.

Define $2^n_0 = n$ and $2^n_{k+1} = 2^{2^n_k}$.

Theorem 7 *1. There exists a constant c such that for all k and n, if S is a schedule of size $\leq n$ for an $\mathbf{A}_k$-match, then every Oracle can be beaten by Us using at most 2^n_{ck} queries.*

2. There exists a constant $\varepsilon > 0$ such that for all k and n, there exists a schedule S of size $\leq n$ for an $\mathbf{A}_k$-match and a strategy for Oracle that cannot be beaten using less than $2^n_{\varepsilon k}$ queries.

Proof. 1. Let S be a schedule for an $\mathbf{A}_k$-match of size n. Introduce function symbols for every move of Alice and predicates for every game. Now we can consider each move of a player as a natural number, so it will be represented by a single variable. In particular, the predicates for games will be at most k-ary.

Let $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$ be the formula that expresses that Bob can win the match determined by the schedule S. The formula Θ has the form

$$\bigwedge_{\ell=1}^{q} (\bigwedge_j \phi^0_{\ell j} \vee \bigwedge_k \neg\phi^1_{\ell k}),$$

as described in Section 3, with a simpler structure of terms: instead of blocks of terms and variables, there are single variables and single terms; the terms have the form $f(y_{p_1}, y_{p_2}, \ldots)$ where f is the function symbol chosen for the particular place in the schedule S. Thus $\phi^0_{\ell j}$ has the form $\phi_\ell(y_{i_1}, s_2, y_{i_3}, s_4 \ldots)$ with terms s_2, s_4 of the form above, and $\phi^1_{\ell k}$ has the dual structure.

We shall show that it has an LK proof of exponential size (in n) with formulas with at most $k+1$ quantifier alterations.[2] For every string of zeros and ones of length q, derive the sequent

$$\ldots, \phi^0_{\ell j}, \ldots, \neg\phi^1_{\ell' k}, \ldots \longrightarrow \Theta,$$

where we pick one conjunction from each disjunction $\bigwedge_j \phi^0_{\ell j} \vee \bigwedge_k \neg\phi^1_{\ell k}$ and put all its terms to the antecedent. (Thus in the sequent above we have decided to put all positive formulas $\phi^0_{\ell j}$ from the disjunction with index ℓ and all negative formulas $\neg\phi^1_{\ell' k}$ from the disjunction with index ℓ'.) Now we shall introduce quantifiers. First we introduce existential quantifiers to the formula in the consequent

$$\Theta \longrightarrow \exists y_1 \ldots \exists y_m \Theta.$$

Then we introduce existential and universal quantifiers (for the moves of Alice and Bob, respectively) in the antecedent. We have to introduce them in the reverse order with respect to the order of the moves, and we must always contract the same formulas. Thus we get

$$\ldots, \forall x_{\ell 1} \exists x_{\ell 2} \ldots \phi_\ell(x_{\ell 1}, x_{\ell 2}, \ldots), \ldots, \exists x_{\ell' 1} \forall x_{\ell' 2} \ldots \neg\phi_{\ell'}(x_{\ell' 1}, x_{\ell' 2}, \ldots), \ldots \longrightarrow \exists y_1 \ldots \exists y_m \Theta.$$

From these 2^q sequents we obtain, by applying left $\vee$-introduction, the sequent

$$\ldots, \forall x_{\ell 1} \exists x_{\ell 2} \ldots \phi_\ell(x_{\ell 1}, x_{\ell 2}, \ldots) \vee \exists x_{\ell 1} \forall x_{\ell 2} \ldots \neg\phi_\ell(x_{\ell 1}, x_{\ell 2}, \ldots), \ldots \longrightarrow \exists y_1 \ldots \exists y_m \Theta.$$

The antecedent consists of logically valid sentences that have polynomial size proofs. Using these proofs and cuts we get $\longrightarrow \exists y_1 \ldots \exists y_m \Theta$.

Thus we have an exponential size proof of $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$. Using the the constructive proof of Herbrand's theorem we obtain a tautological disjunction

$$\bigvee_i \Theta(t_{i1}, \ldots, t_{im}) \tag{6}$$

[2] One can also get a polynomial size proof with formulas with at most $2k$ quantifiers.

with at most 2^n_{ck} formulas, for some constant c. The bound on the number of formulas in the disjunction follows from the bounds on cut-elimination (see Chapter 1 of [2]), since the middle sequent in the Middle Sequent Theorem is essentially such a disjunction.

We also need to bound the size of terms involved. One can inspect the proof of the upper bound and show that the size of terms is upper-bounded by the same type of function.

One can also use the following argument. Given a tautological disjunction of the form (6) we shall ask what is the string of terms $\bar{s}$ that contains smallest possible terms that preserve the logical validity and the intended term structure? Logical validity is preserved if pairs of equal atomic formulas in (6) remain equal after substituting terms s_{ij} for terms t_{ij}, which only requires substituting same terms for same terms. The intended term structure means that the terms at the positions of Alice's moves must must have the form

$$f(t_{i_1 j_1}, \ldots, t_{i_r j_r}), \tag{7}$$

where f the function symbol assigned to the particular move of Alice and terms $t_{i_1 j_1}, \ldots, t_{i_r j_r}$ correspond to the moves of Bob which precede the move of Alice. Such a string of minimal terms $\bar{s}$ is the most general unifier of the unification problem defined as follows. Pick a variable z_{ij} for every term t_{ij} of (6). The equations are of two types. First, for every pair $t_{ij}, t_{i'j'}$ of equal terms we have the equation

$$z_{ij} = z_{i'j'}.$$

Second, for every Alice's move represented by the term t_{ij} of the form (7), we have an equation

$$z_{ij} = f(z_{i_1 j_1}, \ldots, z_{i_r j_r}).$$

It is well-known that the size of terms in the most general unifier is at most exponential in the size of the unification problem. Hence the size of terms in the most general unifier of our problem is at most $2^n_{c'k}$, for some constant c'.

To finish the proof of part 1., we shall show that the number of queries that We need is bounded by the size of the expression (6). Indeed, if there are free variables in the terms, we substitute zero everywhere. Then We evaluate terms by querying Oracle, which will decide the numeric values of expressions $f(n_1, \ldots, n_r)$ for given numbers $n_1, \ldots, n_r$. Finally, We ask Oracle about the outcome of the games in the matches, ie., it will decide if the predicate assigned to a particular game is true for a particular string of integers. Since (6) is a tautology, at least one term in the disjunction will be true. The values of the terms in that disjunction are Bob's winning moves in the match.

2. We shall use the results of Statman and Orevkov [11, 16] (see also a presentation in [13]) that for every k and n there exists an existential formula Ψ such that

1. Ψ has an LK proof d with formulas with at most k quantifiers and size bounded by a polynomial in n,
2. every Herbrand disjunction proving Ψ has at least $2^n_{\delta k}$ terms, for some constant $\delta > 0$.

Let Ψ be $\exists x_1 \ldots \exists x_r \psi(x_1, \ldots, x_r)$ and let a polynomial size proof d of it be given. By Lemma 2, we have a propositional proof d' of

$$\Theta(y_1, \ldots, y_m) \rightarrow \bigvee_{i=1}^{p} \psi(t_{i1}, \ldots, t_{ir}),$$

where Θ is as in the lemma. The size of d' bounded by a polynomial in the size of d, so it is polynomially bounded in the size of Ψ. If a propositional tautology

$$\bigvee_{j=1}^{L} \Theta(s_{j1}, \ldots, s_{jm})$$

is a Herbrand proof of $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$, then the propositional tautology

$$\bigvee_{j=1}^{L} \bigvee_{i=1}^{p} \psi(t_{i1}[\bar{y}/\bar{s}], \ldots, t_{ir}[\bar{y}/\bar{s}])$$

is a Herbrand proof of Ψ. Because of the lower bound on the Herbrand proofs of Ψ, every Herbrand proof of $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$ has at least $L \geq 2^n_{\delta k}/p$ terms in the disjunction. Thus $L \geq 2^n_{\delta' k}$, for some constant $\delta' > 0$.

Let $\mathcal{T}_M$ be the set of terms of depth $\leq M$ in a language with M variables and m function symbols, with each symbol at most m-ary. In Polish notation such terms are represented by strings of length $\leq m^{M+1}$ in an alphabet of size $m + M$. Thus

$$|\mathcal{T}_M| \leq (m + M)^{m^{M+1}}$$

is an upper bound on the size of $\mathcal{T}_M$. Then the number of m-tuples of such terms is at most

$$(m + M)^{m^{M+2}} \leq 2^{2^{mM}},$$

if m and M are sufficiently large.

Let L denote the least number of terms in a Herbrand proof of $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$. Suppose that $2^{2^{mM}} < L$. Let

$$\bigvee_{j=1}^{K} \Theta(s_{j1}, \ldots, s_{jm}) \tag{8}$$

be a disjunction in which all m tuples of terms of $\mathcal{T}_M$ are used. So $K < L$. Pick a truth assignment to the atomic formulas of this formula which makes it false, ie., each term in the disjunction is false. We shall describe Oracle's strategy that enables it not to loose when at most M queries are asked.

1. Each time Oracle is asked to tell the number that represents Alice's move it picks a number that has not been used yet (unless We ask the same question).

2. Oracle associates the numbers used in the queries with terms as follows. When it says that Alice would play number u after Bob played $u_1, \ldots, u_h$ and these numbers have been associated with terms $t_1, \ldots, t_h$ and f is the function symbol assigned to Alice's position in the schedule following these moves, then Oracle associates u with $f(u_1, \ldots, u_h)$. If u is asked in some query where it represents Bob's move and it is not associated with a term yet, then Oracle assigns next free variable to u.

3. To answer a query about who wins in some final position in some game, Oracle uses the associated terms and the truth values chosen for that atomic formula.

The first two conditions ensure that the correspondence between the played numbers and the terms is one-to-one, therefore the strategy is consistent. The use of truth values that make the disjunction (8) false ensures that We cannot find a winning position for Bob if we ask at most M queries.

Now we shall use the lower bound on Herbrand's proofs of $\exists y_1 \ldots \exists y_m \Theta(y_1, \ldots, y_m)$. We know that $L \geq 2^n_{\delta' k}$, hence if M is maximal such that $2^{2^{mM}} < L$, we have $M \geq 2^n_{\varepsilon k}$ for some $\varepsilon > 0$ (recall that m is polynomial in n).

∎

It seems likely that by more careful computation one could improve the lower bound to 2^n_{k-c} and the upper bound to 2^n_{k+c} for some constant c. We conjecture that the true value is asymptotically $2^{n^c}_{k-1}$, for some constant $c > 0$.

6 How to prove noncoservativity for relativized theories?

Theorem 7 shows that the principles $\mathbf{A}^\infty_k$ form an infinite hierarchy if we measure the strength of a principle by the number of queries needed to solve the search problem. In particular we know that if the gap between k and ℓ, $k < \ell$, is sufficiently large, then for some $r < s$, $\mathbf{A}^\infty_k$ can be solved using 2^n_r queries and $\mathbf{A}^\infty_\ell$ cannot be solved using less than 2^n_s queries for schedules of size n. Notice that 2^n_s is exponentially larger than 2^n_r. This suggests that $\mathbf{A}^\infty_\ell$ cannot be reduced to $\mathbf{A}^\infty_k$ using a polynomial number of queries, but this statement does not have a precise meaning unless we specify the type of reduction. One way to make it precise is as follows.

Given a schedule S of size n for an $\mathbf{A}_\ell$-match, we can write a first order formula that describes it. A polynomial reduction of $\mathbf{A}^\infty_\ell$ to $\mathbf{A}^\infty_k$ is a procedure that from S constructs polynomially many schedules S_i of polynomial size for $\mathbf{A}_k$-matches and for each S_i the games and Alice's strategy are defined by first order formulas of polynomial size such that from solutions to the search problem of these matches and using polynomially many queries to the oracle one can find a solution to the search problem for the schedule S.

It is very tempting to try to use this separation to separate $\forall\Sigma^b_1$ consequences of $T^k_2[R]$ from $T^\ell_2[R]$. If that were possible, namely, if for every k there were $\ell > k$ such that

$\forall\Sigma^b_1(T^k_2[R]) \neq \forall\Sigma^b_1(T^\ell_2[R])$, then we would actually get $\forall\Sigma^b_1(T^k_2[R]) \neq \forall\Sigma^b_1(T^{k+1}_2[R])$ for all $k = 1, 2, \ldots$. This follows from a recent result of Skelley and Thapen [15].

The separation of $\mathbf{A}^\infty_k$ frm $\mathbf{A}^\infty_\ell$ can be used to show that, relative to an oracle, the search problem $S(\mathbf{A}_\ell)$ cannot be reduced to $S(\mathbf{A}_k)$ *using certain restricted reductions*, namely, reductions in which the strategies and the games of the $\mathbf{A}_k$-matches are defined from the strategy and games of the $\mathbf{A}_\ell$-match using first order formulas. The idea is the same as in the proof of part 2 of Theorem 7, but instead of assigning arbitrary numbers to terms, we would assign only numbers from an exponential range, or equivalently, bit strings of polynomial length. Since we are only considering reduction that use polynomially many queries, we can do it in a one-to-one manner.

One can also show that $S(\mathbf{A}_\ell)$, relative to an oracle, cannot be reduced to $S(\mathbf{A}_k)$ by reductions which use $\mathbf{A}_k$-matches defined without the oracle. (In such a case k can be arbitrary, since a solution to a nonrelativized $\mathbf{A}_k$-match does not say anything about the oracle.) Thus the difficult case, which are not able to cope with, is when the $\mathbf{A}_k$-matches are defined using both the oracle and the bit structure.

References

[1] S.R. Buss, *Bounded Arithmetic,* Bibliopolis, 1986.

[2] S.R. Buss, An Introduction to Proof Theory, in *Handbook of Proof Theory*, Elsevier, 1998, pp.1-78.

[3] S.R. Buss and J. Krajíček, An application of boolean complexity to separation problems in bounded arithmetic, *Proc. of the London Math. Soc.* 69(3), 1994, pp.1-21.

[4] J. Hanika, Herbrandizing search problems in Bounded Arithmetic, *Math. Log. Q.*, 50(6), 2004, pp.577-586.

[5] S. Hetzl, Describing proofs by short tautologies, prerpint, 2007.

[6] J. Krajíček, No counter-example interpretation and interactive computations, in *Logic from Computer Science,* ed. Y.N. Moschovakis, Springer-Verlag 1992, pp.287-293.

[7] J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory,* Encyclopedia of Mathematics and its Applications 60, Cambridge Univ. Press, 1995.

[8] J. Krajíček and P. Pudlák, Quantified propositional calculi and fragments of Bounded Arithmetic, *Zeitschrift f. math. Logik und Grundlagen d. Math. 36,* 1990, pp.29-46.

[9] J. Krajíček, P. Pudlák and G. Takeuti, Bounded Arithmetic and the Polynomial Hierarchy, *Annals of Pure and Applied Logic 52,* 1991, pp.143-153.

[10] J. Krajíček and G. Takeuti, On induction-free provability, *Annals of Mathematics and Artificial Inteligence 6,* 1992, pp.107-126.

[11] V. P. Orevkov, Lower bounds on the increase in complexity of deductions in cut elimination, *Soviet Mathematics 20.* Original Russian version in *Zapiski Nauchnych Seminarov LOMI 88*, (1979), pp.137-162.

[12] P. Pudlák, Some relations between subsystems of arithmetic and the complexity of computations, in *Logic from Computer Science,* ed. Y.N. Moschovakis, Springer-Verlag 1992, pp.308-317.

[13] P. Pudlák, The lengths of proofs, in *Handbook of Proof Theory*, Elsevier, 1998, pp.547-637.

[14] P. Pudlák, Consistency and games—in search of new combinatorial principles, in *Proc. Logic Colloquium'03,* Helsinki, eds. V. Stoltenberg-Hansen and J. Vaananen, Assoc. for Symbolic Logic 2006, pp.244-281.

[15] A. Skelley and N. Thapen, The provably total search problems of Bounded Arithmetic, preprint, preprint, 2007.

[16] R. Statman, Proof search and speed-up in the predicate calculus, *Annals of Math. Logic 15,* (1978), pp.225-287.