Alternating minima and maxima, Nash equilibria and Bounded Arithmetic

Pavel Pudlák and Neil Thapen

November 27, 2009

Abstract

We show that the least number principle for strict Σ^b_k formulas can be characterized by the existence of alternating minima and maxima of length k. We show simple prenex forms of these formulas whose herbrandizations (by polynomial time functions) are $\forall\hat{\Sigma}^b_1$ formulas that characterize $\forall\hat{\Sigma}^b_1$ theorems of the levels T^k_2 of the Bounded Arithmetic Hierarchy, and we derive from this another characterization, in terms of a search problem about finding pure Nash equilibria in k-turn games.

Introduction

One of the main objects in proof-complexity is the Bounded Arithmetic Hierarchy. This is the proof-complexity counterpart of the Polynomial Hierarchy which is studied in computational complexity. The theories in the Bounded Arithmetic Hierarchy are essentially Peano Arithmetic with induction limited to bounded formulas with k alternations of bounded quantifiers, where k is the level in the hierarchy. More precisely, in order to define T^k_2, the theory on the k-th level, one chooses a suitable set of bounded formulas $\hat{\Sigma}^b_k$ that define precisely the sets in the complexity class Σ^p_k. The theory T^k_2 is axiomatized by a finite set of basic axioms and the induction schema for $\hat{\Sigma}^b_k$ formulas. It is well-known that induction can be replaced by various other principles, in particular by the least number principle.

In this paper we will introduce another principle. Our principle says that, for a polynomial time computable function $v(p, x_1, \ldots, x_k)$, for each p there exists

$$\min_{x_1} \max_{x_2} \min_{x_3} \ldots v(p, x_1, \ldots, x_k),$$

where the minima and maxima are over $x_1, \ldots, x_k \leq p$. This simple result is proved in order to derive another one, which is the essence of this paper: we give new characterizations of the $\forall\hat{\Sigma}^b_1$ sentences (these are, essentially, sentences with a universal quantifier followed by

Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, CZ-115 67 Praha 1, {pudlak, thapen}@math.cas.cz. Partially supported by Institutional Research Plan AV0Z10190503 and grant IAA100190902 of GA AV ČR and by a grant from the John Templeton Foundation.

an existential bounded quantifier) that are provable in T_2^k, for $k = 1, 2, \ldots$. The alternating minima and maxima serve not only to prove these characterizations, but also to help us to more fully understand the meaning of the sentences used in these characterizations.

The study of provable $\forall\hat{\Sigma}_1^b$ sentences is an active research area in proof complexity. These sentences are interesting for two reasons. In proof complexity we associate theories with classes defined in computational complexity by postulating induction for classes of formulas that define these complexity classes. If S and T are theories associated with some natural complexity classes C and D (respectively) and it is conjectured that $C \neq D$, we also conjecture that the (sets of theorems of) S and T are different. It also seems likely that in such a case S and T should differ in their provable $\forall\hat{\Sigma}_1^b$ sentences. We do not know if T_2^{k+1} is strictly stronger than T_2^k, but one can prove this using the assumption that the Polynomial Hierarchy is strictly increasing, and one can also show that relativized versions of these theories are different. However, the separations obtained in these results are by sentences of increasing complexity. Whether one can improve these separations to $\forall\Sigma_1^b$ sentences is still an open problem.

The second reason for studying these sentences is that the set of all true $\forall\hat{\Sigma}_1^b$ sentences defines exactly the class of total polynomial search problems, denoted **TFNP** (standing for *'total functional NP'*). Various subclasses of **TFNP** have been studied in computational complexity theory. Proof complexity provides tools for showing separations of the relativized versions of these classes. Proof complexity is also a source of new subclasses of **TFNP**.

By a characterization of the $\forall\hat{\Sigma}_1^b$ sentences provable in the theories T_2^k we mean an explicitly defined set of $\forall\hat{\Sigma}_1^b$ sentences that are provable in T_2^k and from which all $\forall\hat{\Sigma}_1^b$ sentences are derivable over the basis theory T_2^0 (in fact our characterizations are in terms of a slightly stronger notion, search problem reducibility, which we explain in Section 2 below). Such characterizations for all k were obtained fairly recently [13, 17, 15, 3]. Previously they were known only for $k = 0, 1$ and 2 [5, 9]. While we do not have a clue how to prove conditional separations, it seems that the standard method of proof complexity should work for relativized separations. This method is based on translating the sentences into sequences of tautologies and proving lower bounds on the lengths of proofs of these tautologies. Unfortunately, the application of this method is hindered by the extreme complexity of the combinatorial problems that have to be solved. Therefore, researchers are looking for characterizations by simpler sentences than the known ones, which is also the main aim of this paper.

We will present here two new characterizations of the $\forall\hat{\Sigma}_1^b$ sentences provable in the theories T_2^k. In the first one our sentences are similar to those in [17, 3], but simpler. Their simplicity may help to prove relativized separations. We obtain the sentences by first writing the principle of alternating minima and maxima in a suitable prenex form and then taking a herbrandization by polynomial time computable functions. This means that we eliminate the universal quantifiers of the prenex formula by introducing function symbols, as in Herbrand's Theorem, and then use this as a schema in which the new function symbols represent polynomial time computable functions. We discovered these sentences several years ago and conjectured that they characterize the $\forall\hat{\Sigma}_1^b$ sentences provable in the theories T_2^k

[16]. But we only recently realized that there is a reduction of the Game Induction Principle of [17] to our sentences.

The second characterization is as a problem about finding equilibrium strategies for a game. It was recently shown [6, 7] that the general problem of finding a mixed Nash equilibrium is complete for the search problem class **PPAD**, and there is active ongoing research into the computational complexity of game theory. In this setting, the standard way to present a game is in *strategic form*, where essentially each player has only one move and all players move simultaneously, and the standard way to input such a game to a machine is as an explicitly given table of payoffs. In contrast, our games are in *sequential form*, where we think of the (two) players as taking turns to move, and they are *zero-sum*, with the players having opposite payoffs; it is straightforward to show that a pure Nash equilibrium always exists (so we do not have to consider probabilistic strategies). Furthermore our payoff functions are given *succinctly*, by a polynomial time function rather than a table – having such a table as input would put the problem trivially into polynomial time. Similar kinds of games are considered in [1] where it is shown that many decision problems about such games are PSPACE-complete. The general question of how hard it is to find pure equilibria in games where they are known to exist is raised in [8], where it is shown that for *congestion games* this problem is complete for the class **PLS** (however the setting there is different, and this seems to be unconnected to our results about **PLS** in this paper).

For our search problems to be in **TFNP**, we also need to weaken the definition of a Nash equilibrium, in what we feel is a natural way. The usual definition of a pure equilibrium is a pair of strategies for players A and B, such that neither player can improve his payoff by unilaterally switching to a new strategy. We will weaken this by adding the condition that any new strategy must be derivable from the old strategies, considered as oracles, by a polynomial time algorithm. We show that the existence of such equilibria in k-turn games characterizes the $\forall\hat{\Sigma}^b_1$ sentences provable in the theory T^k_2.

1 The Bounded Arithmetic Hierarchy

The theories T^k_2, for $k \geq 0$ were defined by Buss [4] (although our formalization is slightly different from his; see the last paragraphs of this section). They are formalized in the language with primitive symbols $0, 1, +, \times, |x|, \#, \lfloor x/2 \rfloor, \leq$. The intended interpretations of $|x|$ is $\lceil \log_2(x+1) \rceil$ (which is the length of the binary expansion of x, if $x > 0$); the interpretation of $x\#y$ is $2^{|x|\cdot|y|}$; and the interpretations of the other symbols are standard. The richer language is needed because the theories have restricted induction schemes. In particular, the function $\#$ enables us to construct, from a number x, the number $x\#(x\#\ldots(x\#x)\ldots)$ (with x occurring k-times) whose length is equal to the length of x raised to the kth power. This is needed for formalizing polynomial time computations.

The theories T^k_2 are axiomatized by a finite set of axioms that fixes the interpretation of the basic notions and by the usual scheme of induction for $\hat{\Sigma}^b_k$ formulas. This class of formulas is defined as follows. First one defines bounded formulas in the usual way. *Sharply bounded quantifiers* are defined by the condition that the outermost term is $|\ldots|$; thus they have forms

$\forall x \leq |t|$ and $\exists x \leq |t|$, where t is a term. Formulas with only sharply bounded quantifiers are called *sharply bounded formulas*. A $\hat{\Sigma}^b_k$ formula consist of at most k alternations of bounded quantifiers, with the first one existential, followed by a sharply bounded formula.

In the theory T^0_2, as defined by Buss, the induction scheme is restricted to formulas with only sharply bounded quantifiers. Since this theory is very weak, Jeřábek has proposed extending the language by the function $\lfloor x/2^y \rfloor$ and a finite number of axioms fixing its interpretation [10]. In the resulting version of the theory T^0_2 it is possible to define polynomial time computations by $\hat{\Delta}^b_1$ formulas. We will use this theory as the basis theory for our results.

Terms of these theories do not suffice to define all polynomial time computable functions (even with the function $\lfloor x/2^y \rfloor$) and sharply bounded formulas do not suffice to define all polynomial time sets and relations. Therefore we shall allow the introduction of a new function symbol representing a polynomial time computable function whenever it has a $\hat{\Delta}^b_1$ definition for which T^0_2 proves that it is polynomial time computable. We shall use a similar convention about polynomial time computable relations. In this richer language we do not need to use sharply bounded quantifiers any more. Thus we may take the class $\hat{\Sigma}^b_k$ to consist of formulas with a quantifier-free part built from symbols for polynomial time functions and predicates, prefixed by some bounded quantifiers. If the defined function symbols and predicates are eliminated from such a formula by substituting their definitions, we obtain an equivalent $\hat{\Sigma}^b_k$ formula.

Buss's original formalization in [4] is in terms of classes Σ^b_k in which sharply bounded quantifiers can appear anywhere in a formula, without increasing its complexity; the $\hat{\Sigma}^b_k$ formulas then correspond to *strict* Σ^b_k formulas. However the strength of the theories T^k_2 is not changed by restricting induction to strict formulas. We note that our characterization of the provable $\forall\hat{\Sigma}^b_1$ sentences of T^k_2 also gives a characterization of the provable $\forall\Sigma^b_1$ sentences, if we strengthen our basis theory from T^0_2 to Buss's theory S^1_2 in which every Σ^b_1 formula can be shown to be equivalent to a $\hat{\Sigma}^b_1$ formula.

2 Polynomial search problems

Definition 1 *A total polynomial search problem is given by a relation R such that*

1. $R(x, y) \in \mathbf{P}$;
2. *there is a polynomial p such that $R(x, y)$ implies $|y| \leq p(|x|)$;*
3. $\forall x \exists y R(x, y)$.

The problem is: given input x, find y such that $R(x, y)$.
The class of all total polynomial search problems is denoted by **TFNP**.

Definition 2 *Let S_i be a search problem determined by $R_i(x, y)$. Then S_1 is* polynomially many-one reducible *to S_2 if there exist polynomial time computable functions f and g such that given x, f computes some string $f(x) = x'$ such that if $R_2(x', y')$ for some y', then $R_1(x, g(x, y'))$.*

Various classes of **TFNP** problems closed under polynomial reductions have been studied and several separations of relativized classes have been shown (see e.g. [2]).

Clearly, a **TFNP** problem is associated with a true $\forall\hat{\Sigma}^b_1$ sentence (the universal closure of a $\hat{\Sigma}^b_1$ formula) and, vice versa, every true $\forall\hat{\Sigma}^b_1$ formula determines a **TFNP** problem. Furthermore the definition of many-one reducibility of a search problem S_1 to a search problem S_2 can be read as a strong (skolemized) version of logical implication of the sentence for S_2 from the sentence for S_1. Hence our goal will be to show that a scheme Γ characterizes the set of $\forall\hat{\Sigma}^b_1$ consequences of a theory over T^0_2 in a strong way, by explicitly showing how each search problem for a sentence in the set can be reduced to a search problem for a sentence in Γ by a many-one reduction that can be formalized in T^0_2.

It has been proved that all polynomial search problems for which the totality condition 3. is provable in T^0_2 are computable in polynomial time (this is essentially Buss's result from [4]). Wilkie showed (reported in [12]) that if the totality is provable in T^0_2 extended by a surjective version of the weak pigeonhole principle, then the search problem can be solved in probabilistic polynomial time. The first characterization of $\forall\hat{\Sigma}^b_1$ sentences provable at a level of the hierarchy above T^0_2 was due to Buss and Krajíček [5]. They proved that the provably total polynomial search problems in T^1_2 are polynomially reducible to problems from the class **PLS** (standing for *polynomial local search* [11]), and used this to give a relativized separation of T^1_2 and T^2_2 by a $\forall\hat{\Sigma}^b_1$ sentence. A simplified version of the definition of polynomial local search is given in Section 6. It is a special case of the polynomial search problems which we will use to characterize the $\forall\hat{\Sigma}^b_1$ sentences provable in theories T^k_2. For this reason we call our principles and search problems *Generalized Polynomial Local Search*.

3 Some useful sentences

In this section we present some sentences equivalent to the existence of the least number satisfying a certain k-quantifier formula. In Section 5 we will use these sentences to state corresponding axiom schemes and show that they axiomatize the theories T^k_2.

The equivalences that we are going to prove can be proved in a very weak theory. Thus *in this section we will only use pure logic and the assumption that $\leq$ is a discrete linear ordering and $v(p, x_1, \ldots, x_k)$ is an arbitrary function of $k+1$ variables.* The condition that $\leq$ is discrete means that

$$\forall x(\exists y(y < x) \rightarrow \exists x^-(x^- < x \wedge \forall y(y < x \rightarrow y \leq x^-))),$$

and the dual. We shall denote the predecessor (successor) of x, if it exists, by x^- (x^+).

The meaning of the expression

$$z = \min_{x_1} \max_{x_2} \min_{x_3} \ldots v(p, x_1, \ldots, x_k),$$

with k mins and maxs, is clear when all the minima and maxima exist. Since we will deal with situations in which some maxima and minima are not defined, we have to be more careful when using such expressions.

Therefore, assuming k is even (and similarly for k odd), we will use such an expression only if the existence of all minima and maxima is guaranteed for all suffixes

$$\max_{x_i} \ldots \min_{x_{k-1}} \max_{x_k} v(p, x_1, \ldots, x_k)$$

for all $p, x_1, \ldots, x_{i-1}$ and all i odd, and

$$\min_{x_j} \ldots \min_{x_{k-1}} \max_{x_k} v(p, x_1, \ldots, x_k)$$

for all $p, x_1, \ldots, x_{j-1}$ and all j even. Thus, in particular,

$$\exists z \ (z = \min_{x_1} \max_{x_2} \min_{x_3} \ldots v(p, x_1, \ldots, x_k))$$

is an abbreviation for the formula where the existence is stated for all suffixes as above.

The variable p serves only as a parameter, therefore in the rest of this section it will be omitted. Note that the results below have duals in which max and min are switched and $\leq$ is reversed. We shall use the dual versions without comment when needed.

Theorem 3.1 *The following two sentences are equivalent:*

$$\exists u \ (u = \min\{w; \ \exists x_1 \forall x_2 \exists x_3 \ldots v(x_1, \ldots, x_k) \leq w\}), \tag{1}$$

$$\exists x_1 \forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \ldots (v(x_1, \ldots, x_k) \leq v(y_1, \ldots, y_k)). \tag{2}$$

(The last two quantifiers in (2) are $\exists x_k \forall y_k$ if k is odd, and $\exists y_k \forall x_k$ if k is even.)

Furthermore, if for all x_1, $\max_{x_2} \min_{x_3} \ldots v(x_1, \ldots, x_k)$ exists, then (3) and (2) are equivalent to

$$\exists u \ (u = \min_{x_1} \max_{x_2} \min_{x_3} \ldots v(x_1, \ldots, x_k)). \tag{3}$$

The diagram below illustrates the order of quantifiers in (2).

$$\begin{array}{ccccccc} \exists x_1 & & \forall x_2 & \rightarrow & \exists x_3 & & \ldots \\ \downarrow & & \uparrow & & \downarrow & & \\ \forall y_1 & \rightarrow & \exists y_2 & & \forall y_3 & \rightarrow & \ldots \end{array} \tag{4}$$

We will prove the theorem by a sequence of lemmas.

Lemma 3.2 *Sentence (1) is equivalent to the following sentence (5):*

$$\exists w \ [\exists x_1 \forall x_2 \exists x_3 \ldots (v(x_1, \ldots, x_k) \leq w) \wedge \forall y_1 \exists y_2 \forall y_3 \ldots (v(y_1, \ldots, y_k) \geq w)]. \tag{5}$$

Proof. Sentence (5) is clearly as strong as the existence of the minimum. For the opposite direction, we shall use the discreteness of $\leq$. Let $w = \min\{w; \ \exists x_1 \forall x_2 \exists x_3 \ldots v(x_1, \ldots, x_k) \leq w\}$. The first part of (5) is immediate. To get the second part, observe that w satisfies:

$$\forall u \ (u < w \rightarrow \forall y_1 \exists y_2 \forall y_3 \ldots (v(y_1, \ldots, y_k) > u)).$$

If there is no $u < w$ then the second part is clear. Otherwise $w^- < w$, so

$$\forall y_1 \exists y_2 \forall y_3 \ldots (v(y_1, \ldots, y_k) > w^-),$$

whence

$$\forall y_1 \exists y_2 \forall y_3 \ldots (v(y_1, \ldots, y_k) \geq w).$$

Notice that we have shown that if w is the minimum in (1), then it satisfies the inequalities in (5). ∎

To prove (1)⇔(3) we prove the following slightly stronger lemma.

Lemma 3.3 *Suppose that for all x_1, $\max_{x_2} \min_{x_3} \ldots v(x_1, \ldots, x_k)$ exists. If furthermore one of the two numbers defined by the expressions in the following equality exists, then the other exists too and they are equal:*

$$\min_{x_1} \max_{x_2} \min_{x_3} \ldots v(x_1, \ldots, x_k) = \min\{w;\ \exists x_1 \forall x_2 \exists x_3 \ldots v(x_1, \ldots, x_k) \leq w\}.$$

Proof. We shall use the following easy fact: if $\min X$ exists and

$$\forall x \in X \exists y \in Y (y \leq x) \ \wedge\ \forall y \in Y \exists x \in X (x \leq y), \tag{6}$$

then $\min Y$ exists and $\min X = \min Y$.

We prove the lemma by induction on k. The base case $k = 1$ is true by definition.

By induction (applied to the dual statement), we can assume that for every x_1

$$\max_{x_2} \min_{x_3} \ldots v(x_1, \ldots, x_k) = \max\{w;\ \exists x_2 \forall x_3 \ldots v(x_1, \ldots, x_k) \geq w\}.$$

Thus we need to prove that if one of the two numbers defined by the expressions in the following equality exists, then the other exists too and they are equal:

$$\min_{x_1} \max\{w;\ \exists x_2 \forall x_3 \ldots v(x_1, \ldots, x_k) \geq w\} = \min\{w;\ \exists x_1 \forall x_2 \exists x_3 \ldots v(x_1, \ldots, x_k) \leq w\}.$$

Thus it suffices to prove (6) for

$$X = \{w;\ \exists x_1 \forall x_2 \exists x_3 \ldots v(x_1, \ldots, x_k) \leq w\}$$
$$\text{and} \quad Y = \{u;\ \exists x_1\ u = \max\{z;\ \exists x_2 \forall x_3 \ldots v(x_1, \ldots, x_k) \geq z\}\}.$$

To prove the first part of (6), let $w \in X$ and let b be such that

$$\forall x_2 \exists x_3 \ldots v(b, x_2, \ldots, x_k) \leq w.$$

Let $u = \max\{u;\ \exists x_2 \forall x_3 \ldots v(b, x_2, \ldots, x_k) \geq u\} \in Y$. If $u > w$, then

$$\exists x_2 \forall x_3 \ldots v(b, x_2, \ldots, x_k) > w,$$

which is in contradiction with the condition above. Thus $u \leq w$.

For the second part of (6), let $u \in Y$, so

$$u = \max\{u;\ \exists x_2 \forall x_3 \ldots v(a, x_2, \ldots, x_k) \geq u\}$$

for some a. As we observed in the proof of Lemma 3.2, u satisfies $\forall x_2 \exists x_3 \ldots v(a, x_2, \ldots, x_k) \leq u$. Hence $u \in X$. ∎

Finally for (1)⇔(2) it is sufficient to prove the following.

Lemma 3.4 *Sentence (5) is equivalent to (2).*

Proof. For (5)⇒(2), transform (5) into the following prenex form

$$\exists w \exists x_1 \forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \ldots (v(x_1, \ldots, x_k) \leq w \wedge w \leq v(y_1, \ldots, y_k)),$$

which, clearly, implies (2).

For (2)⇒(5) we shall use induction over k. For $k = 1$, there is nothing to prove, because (2) says that there is a minimum of $v(x_1)$.

Suppose that the theorem is true for $k - 1$. Let (2) be true and a be such that

$$\forall y_1 \exists y_2 \forall x_2 \exists x_3 \forall y_3 \ldots (v(a, x_2, \ldots, x_k) \leq v(y_1, y_2, \ldots, y_k)). \tag{7}$$

Thus we have

$$\exists y_2 \forall x_2 \exists x_3 \forall y_3 \ldots (v(a, x_2, \ldots, x_k) \leq v(a, y_2, \ldots, y_k)).$$

By the (dual of the) induction assumption, this implies

$$\exists w \ [\exists y_2 \forall y_3 \ldots (v(a, y_2, \ldots, y_k) \geq w) \wedge \forall x_2 \exists x_3 \ldots (v(a, x_2, \ldots, x_k) \leq w)].$$

Let c be such a w, i.e., we have (8) and (9) below:

$$\exists y_2 \forall y_3 \ldots (v(a, y_2, \ldots, y_k) \geq c), \tag{8}$$

$$\forall x_2 \exists x_3 \ldots (v(a, x_2, \ldots, x_k) \leq c). \tag{9}$$

We shall show that (10) and (11) below also hold

$$\exists x_1 \forall x_2 \exists x_3 \ldots (v(x_1, \ldots, x_k) \leq c) \tag{10}$$

$$\forall y_1 \exists y_2 \forall y_3 \ldots (v(y_1, \ldots, y_k) \geq c), \tag{11}$$

which will finish the proof.

First, (10) is an immediate consequence of (9). To prove (11) we shall argue by contradiction. Suppose it is false. Take the conjunction of (8), with ys renamed to xs, with the negation of (11)

$$\exists x_2 \forall x_3 \ldots (v(a, x_2, \ldots, x_k) \geq c) \wedge \exists y_1 \forall y_2 \exists y_3 \ldots (v(y_1, \ldots, y_k) < c)$$

and put it into the prenex form

$$\exists y_1 \forall y_2 \exists x_2 \forall x_3 \exists y_3 \ldots (v(a, x_2, \ldots, x_k) \geq c \wedge v(y_1, \ldots, y_k) < c).$$

This is in contradiction with (7). Hence (11) is true. ∎

4 An interpretation in terms of games

We can interpret the concepts introduced above in terms of games. Given a function $v(x_1, x_2, \ldots, x_k)$ of k variables, consider the game G in which two players A and B alternate in choosing values for $x_1, x_2, \ldots, x_k$. After playing these numbers the game ends and A loses $v(x_1, x_2, \ldots, x_k)$ and B gains $v(x_1, x_2, \ldots, x_k)$. Thus the aim of A, who starts, is to minimize the payoff, while B tries to maximize it (we will come back to this game in Section 7). The number

$$w = \min_{x_1} \max_{x_2} \min_{x_3} \ldots v(x_1, x_2, \ldots, x_k),$$

has the properties:

- there exists a strategy for A not to lose more than w;
- there exists a strategy for B to gain at least w.

In particular, the two strategies form an equilibrium.

The existence of such a w (the sentence (3) in Theorem 3.1) in general may be not provable if the theory is too weak.

For sentence (2), consider the game H in which two players C and D play two copies of G simultaneously. C plays as A in the first copy and as B in the second copy. The order of moves is shown in the diagram (4), with C playing as the existential quantifier and D the universal. If $\bar{x}$ and $\bar{y}$ are the moves from the first and second copy, C wins H if $v(\bar{x}) \leq v(\bar{y})$. The sentence (2) expresses that C can always win H; this is true if the value w above exists.

If (2) is true, then C can in particular still win H if D's moves are played according to some fixed strategy S; but now the universal quantifiers for D's moves disappear and the sentence becomes purely existential. This is essentially the principle $\mathbf{GPLS}_k$ considered in Section 6.

Principles based on the idea of two players playing simultaneously several games was considered in [15]. The games considered in that paper had only two possible values, which was the reason why those principles were much more complicated.

5 Schemes axiomatizing T_2^k

The sentences from Section 3 can be used to axiomatize theories in Bounded Arithmetic. In this setting, we will let v range over polynomial time functions and let minima and maxima be defined over the interval $[0, p]$, where p is the parameter.

Theorem 5.1 *For every $k \geq 1$, the theory T_2^k can be axiomatized by the axioms of T_2^0 together with any of the following three schemes:*

S1(k): $\forall p \exists z\ z = \min_{w \leq p}\{w;\ \exists x_1 \leq p \forall x_2 \leq p \exists x_3 \leq p \forall x_4 \leq p \ldots\ v(p, x_1, \ldots, x_k) \leq w\}$;

S2(k): $\forall p \exists x_1 \leq p \forall y_1 \leq p \exists y_2 \leq p \forall x_2 \leq p \ldots\ v(p, x_1, \ldots, x_k) \leq v(p, y_1, \ldots, y_k)$.

S3(k): $\forall p \exists z\ z = \min_{x_1 \le p} \max_{x_2 \le p} \min_{x_3 \le p} \max_{x_4 \le p} \dots\ v(p, x_1, \dots, x_k)$.

Here v denotes a formalization of a polynomial time computable function in T^0_2 such that $v(p, x_1, \dots, x_k) \le p$ for all $p, x_1, \dots, x_k$.

Proof. Note that schemes S1(k) and S2(k) imply S1(j) and S2(j) for all $j \le k$. Hence by Theorem 3.1, the schemes S1(k), S2(k) and S3(k) are equivalent. We will show that S1(k) is equivalent to the least number principle for $\hat{\Sigma}^b_k$ formulas, which is the following scheme

$$\forall p\ (\exists y \le p \exists x_1 \le p \forall x_2 \le p \exists x_3 \le p \forall x_4 \le p \dots \phi(p, y, x_1, \dots, x_k) \rightarrow$$
$$\exists z \le p\ z = \min_{y \le p}\{y;\ \exists x_1 \le p \forall x_2 \le p \exists x_3 \le p \forall x_4 \le p \dots \phi(p, y, x_1, \dots, x_k)\}),$$

for every polynomial time (in T^0_2) predicate ϕ. It is well-known that this axiomatizes T^k_2, cf. [4].

First observe that S1(k) is a special case of the least number principle for $\hat{\Sigma}^b_k$ formulas, giving us one direction of the theorem. For the other direction, let $\phi(p, y, x_1, \dots, x_k)$ be given. We define a polynomial time function $v(p, x_1, \dots, x_k)$ by

$$v(p, (y, x), x_2, \dots, x_k) = \begin{cases} y & \text{if } y \le p \text{ and } \phi(p, y, x, x_2, \dots, x_k), \\ p & \text{otherwise.} \end{cases}$$

Let us write $\Phi(p, y, x)$ for

$$\forall x_2 \le p \exists x_3 \le p \forall x_4 \le p \dots \phi(p, y, x, x_2, \dots, x_k).$$

Then we have, for all $y < p$,

$$\forall x_2 \le p \exists x_3 \le p \forall x_4 \le p \dots\ v(p, (y, x), x_2, \dots, x_k) \le y \ \Leftrightarrow\ \Phi(p, y, x). \tag{12}$$

In order to apply S1(k) we have to modify the scheme a little, by changing the bounded quantifier $\exists x_1 \le p$ to $\exists x_1 \le (p+1)^2$ so that x_1 can code any pair (y, x) with $y, x \le p$. As in other schemes studied in bounded arithmetic, this is an inessential change; one can easily derive the modification from the form stated in the theorem.

Let z be the minimum given by S1(k). We claim that

$$\neg\, \exists y < z \exists x \le p\ \Phi(p, y, x), \tag{13}$$

because if there were such y and x we would obtain

$$\exists x_1 \le (p+1)^2 \forall x_2 \le p \exists x_3 \le p \forall x_4 \le p \dots\ v(p, x_1, \dots, x_k) = y < z$$

from (12) (taking $x_1 = (y, x)$), contradicting the minimality of z. Suppose also that the antecedent of the least number principle is true, that is,

$$\exists y \le p \exists x \le p\ \Phi(p, y, x). \tag{14}$$

Now consider two cases. First, suppose that $z = p$. Then by (14) and (13), p is the least y satisfying $\exists x \le p\ \Phi(p, y, x)$. Second, suppose that $z < p$. Then z satisfies $\exists x \le p\ \Phi(p, z, x)$ by (12), and is the least number satisfying this by (13). ∎

6 Generalized Polynomial Local Search

In this section we shall show that the herbrandization of (2) (more precisely, of the sentences S3 of Theorem 5.1) characterizes the $\forall\hat{\Sigma}^b_1$ theorems of T^k_2. The herbrandization of (2) is:

$$\exists x_1 \exists y_2 \exists x_3 \ldots \; v(x_1, h_2(x_1, y_2), x_3, \ldots) \leq v(h_1(x_1), y_2, h_3(x_1, y_2, x_3), \ldots).$$

We shall call the computational versions of these sentences $\mathbf{GPLS}_k$ problems. Here is a formal definition (with the parameter p explicitly mentioned).

Definition 3 *A* $\mathbf{GPLS}_k$ problem *is defined by* polynomial time *functions v depending on $k+1$ variables and $h_1, \ldots, h_k$ depending on $2, 3, 4, \ldots, k+1$ variables respectively (the first variable is a parameter). An instance of the problem is given by a number a, a value of the parameter. The goal is to find numbers $b_1, c_2, b_3, c_4, \ldots \leq a$, values of $x_1, y_2, x_1, y_2, \ldots$, such that*

$$v(a, b_1, h_2(a, b_1, c_2), b_3, \ldots) \leq v(a, h_1(a, b_1), c_2, h_3(a, b_1, c_2, b_3), \ldots). \tag{15}$$

The formalization of this $\mathbf{GPLS}_k$ *problem in Bounded Arithmetic is the sentence*

$$\forall p \exists x_1 \leq p \exists y_2 \leq p \exists x_3 \leq p \ldots$$
$$v(p, x_1, h_2(p, x_1, y_2), x_3, \ldots) \leq v(p, h_1(p, x_1), y_2, h_3(p, x_1, y_2, x_3), \ldots).$$

The $\mathbf{GPLS}_k$ *scheme is the set of these sentences.*

In particular, if $k = 1$ these problems are special cases of **PLS** problems: v is the cost function, h_1 is the neighborhood function and every $x_1 \leq a$ is a feasible solution; the goal is, for a given parameter a, to find a feasible solution b_1 such that the neighborhood function h_1 does not decrease the cost, i.e., $v(a, b_1) \leq v(a, h_1(a, b_1))$.

Theorem 6.1 *The* $\mathbf{GPLS}_k$ *scheme characterizes over T^0_2 the $\forall\hat{\Sigma}^b_1$ sentences provable in T^k_2, in the strong sense that*

1. *The* $\mathbf{GPLS}_k$ *scheme is provable in T^k_2;*
2. *Every search polynomial search problem provably total in T^k_2 is reducible to a* $\mathbf{GPLS}_k$ *problem, and the reduction can be formalized in T^0_2.*

Proof. For 1., by Theorem 5.1, T^k_2 proves all the sentences of the scheme S3. Since every sentence implies its herbrandizations, T^k_2 also proves the sentences of the $\mathbf{GPLS}_k$ scheme.

For 2., we will reduce the *k-Game Induction principle* $\mathbf{GI}_k$ of [17] to $\mathbf{GPLS}_k$. This is sufficient, since it was proved in [17] that the total polynomial search problems of T^k_2 are reducible to the k-Game Induction principle (considered as a class of search problems), provably in T^0_2.

In the Game Induction principle games with only two values 0 (lose) and 1 (win) are used. Let $G(x_1, \ldots, x_k)$ be a function representing such a game. A *winning strategy* for the first (respectively, second) player is a string of functions $s_1, s_3, \ldots$ $(t_2, t_4, \ldots)$ such that

$$\forall x_2 \forall x_4 \ldots G(s_1(), x_2, s_3(x_2), \ldots) = 1,$$

respectively,

$$\forall x_1 \forall x_3 \ldots G(x_1, t_2(x_1), x_3, t_4(x_1, x_3), \ldots) = 1.$$

A *reduction* of a game G to a game H is a strategy to play G as the first player assuming that we know how to play H as the first player. Formally, it is a string of functions $f_1, \ldots, f_k$ such that

$$\forall x_1 \forall y_2 \forall x_3 \ldots \; H(x_1, f_2(x_1, y_2), x_3, \ldots) \leq G(f_1(x_1), y_2, f_3(x_1, y_2, x_3), \ldots).$$

The principle $\mathbf{GI}_k$ states that it is impossible to have games $G_0, G_1, \ldots, G_a$ and

1. a winning strategy for the first player in G_0,
2. reductions of G_{i+1} to G_i for $i = 0, \ldots, a-1$, and
3. a winning strategy for the second player in G_a.

The principle naturally gives rise to a class $\mathbf{GI}_k$ of search problems by letting the games, strategies and reductions be given by polynomial time functions and bounding all moves by the parameter a.

We describe a reduction of $\mathbf{GI}_k$ to $\mathbf{GPLS}_k$. As we noted above, we can easily prove that w.l.o.g. we can assume that the first argument in v in a $\mathbf{GPLS}_k$ problem is encoding a pair (i, x_1) of numbers $\leq a$. We will also assume that for a given parameter a the value of v is bounded by $2a+1$. Given an instance of $\mathbf{GI}_k$, define an instance of $\mathbf{GPLS}_k$ as follows (we omit the parameter a for the sake of readability):

$$v((i, x_1), x_2, \ldots, x_k) := 2a + 1 - (a+1)g(i, x_1, x_2, \ldots, x_k) - i,$$

$$h_1((i, x_1)) := \begin{cases} (i+1, f_1(i, x_1)) & \text{if } i < a, \\ (0, s_1()) & \text{if } i = a, \end{cases}$$

and for $j = 2, \ldots, k$,

$$h_j((i, x_1), x_2, \ldots, x_j) := \begin{cases} f_j(i, x_1, x_2, \ldots, x_j) & \text{if } i < a, \\ s_j(x_2, x_4, \ldots, x_{j-1}) & \text{if } i = a \text{ and } j \text{ is odd}, \\ t_j(x_1, x_3, \ldots, x_{j-1}) & \text{if } i = a \text{ and } j \text{ is even}. \end{cases}$$

First let us observe that for $i < a$,

$$\begin{aligned} v((i, x_1), x_2, \ldots, x_k) &\leq v((i+1, y_1), y_2, \ldots, y_k) \;\Leftrightarrow \\ -(a+1)g(i, x_1, x_2, \ldots, x_k) - i &\leq -(a+1)g(i+1, y_1, y_2, \ldots, y_k) - (i+1) \;\Leftrightarrow \\ g(i, x_1, x_2, \ldots, x_k) &> g(i+1, y_1, y_2, \ldots, y_k). \end{aligned} \tag{16}$$

Also

$$\begin{aligned} v((a,x_1),x_2,\ldots,x_k) &\leq v((0,y_1),y_2,\ldots,y_k) \;\Leftrightarrow \\ -(a+1)g(a,x_1,x_2,\ldots,x_k)-a &\leq -(a+1)g(0,y_1,y_2,\ldots,y_k) \;\Leftrightarrow \\ g(a,x_1,x_2,\ldots,x_k)=1 \quad &\text{or} \quad g(0,y_1,y_2,\ldots,y_k)=0. \end{aligned} \tag{17}$$

Now suppose that $a,i,b_1,c_2,b_3,c_4,\ldots \leq a$ is a solution of the $\mathbf{GPLS}_k$ problem. Thus

$$v((i,b_1),h_2((i,b_1),c_2),b_3,\ldots) \leq v(h_1((i,b_1)),c_2,h_3((i,b_1),c_2,b_3),\ldots).$$

If $i<a$, then by (16) and the definition of the functions h_j,

$$g(i,b_1,f_2(i,b_1,c_2),b_3,\ldots) > g(i+1,f_1(i,b_1),c_2,f_3(i,b_1,c_2,b_3),\ldots),$$

which shows that the functions $f_j(i,\ldots)$ are not a reduction of G_{i+1} to G_i.

If $i=a$, then by (17) and the definition of the functions h_j,

$$g(a,b_1,t_2(b_1),b_3,\ldots)=1 \quad \text{or} \quad g(0,s_1(),c_2,s_3(c_2),\ldots)=0,$$

which shows that either $t_2,t_4,\ldots$ is not a winning strategy for the second player in G_a, or $s_1,s_3,\ldots$ is not a winning strategy for the first player in G_0.

Finally note that this reduction only uses elementary operations with polynomial time computable functions, hence can be formalized in T^0_2. Thus in T^0_2 the existence of a solution of an instance of $\mathbf{GI}_k$ follows from the existence of a solution of an instance of $\mathbf{GPLS}_k$. ■

We note that we can slightly simplify the formal definition of $\mathbf{GPLS}_k$ problems by assuming that v defines a game in which each move encodes all previous moves. We can force players to only play such moves by punishing the first one to deviate from this rule. Formally, it means that we replace a value function v by another one $\hat{v}$ defined by

$$\hat{v}(a,x_1,(x_1,x_2),(x_1,x_2,x_3),\ldots) := v(a,x_1,x_2,x_3,\ldots),$$

and

$$\hat{v}(a,y_1,y_2,y_3,\ldots)=0 \quad (\text{respectively}, \quad =a),$$

if $y_1=x_1, y_2=(x_1,x_2),\ldots,y_{j-1}=(x_1,x_2,\ldots,x_{j-1})$, where $x_1,x_2,\ldots,x_{j-1}\leq a$, but y_j does not have this form and j is even (respectively, j is odd).

In such games the herbrand functions $h_2,\ldots,h_k$ can formally depend only on two moves (and the parameter). Thus the principle gets the following form:

$\forall p \exists x_1 \leq p \exists y_2 \leq p \exists x_3 \leq p y_4 \leq p \ldots$

$$v(p,x_1,h_2(p,x_1,y_2),x_3,h_4(p,x_3,y_4),\ldots) \leq v(p,h_1(p,x_1),y_2,h_3(p,y_2,x_3),y_4,\ldots).$$

7 Pure Nash equilibria in sequential games

Definition 4 *A* payoff function *is a polynomial time function $v(z, x_1, \ldots, x_k)$, where we think of p as a parameter and $x_1, \ldots, x_k$ as moves in a game, which must be numbers less than or equal to p. A* game *consists of players A and B alternately making moves. A's goal is to minimize the final value of the payoff function and B's is to maximize it.*

The next definition is in the context of a fixed assignment of a value a to the parameter z, defining a particular game.

Definition 5 *A* strategy *S for player A is a tuple $(S_1, S_3, S_5, \ldots)$ of functions telling A which move to make at each of his turns given the history of the game so far, with each S_i a function with domain a^{i-1} and range a. A strategy T for B is defined dually. Strategies should be thought of as oracles, with arguments and values bounded by the parameter; there is no requirement that they are polynomial time computable. We write $v[S, T]$ for the payoff of the game in which A plays with strategy S and B with strategy T.*

The existence of a pure Nash equilibrium (S, T) can now be written as a formula with "second-order" quantifiers over oracles:

$$\exists S, T \, \forall S', T' \, (v[S', T] \geq v[S, T] \, \wedge \, v[S, T'] \leq v[S, T]).$$

That is, A cannot reduce the payoff by unilaterally changing his strategy, and B cannot increase the payoff by unilaterally changing his strategy.

Theorem 7.1 *[14] Every such game has a pure Nash equilibrium.*

Proof. Suppose that k is even. The proof for odd k is similar. We will exhibit two strategies S and T. We define T_k, the last function in B's strategy, by choosing $T_k(x_1, \ldots, x_{k-1})$ to be the number x_k which maximizes the payoff $v(a, x_1, \ldots, x_{k-1}, x_k)$. If there is more than one such x_k, we pick the least one. The last function S_{k-1} in A's strategy is then chosen as the least x_{k-1} which minimizes $v(a, x_1, \ldots, x_{k-2}, x_{k-1}, T_k(x_1, \ldots, x_{k-1}))$. We carry on defining the strategies in this way, backwards from the end of the game, alternating maxima and minima. Notice that these strategies can be given by polynomial time functions with Σ_k^p oracles.

Now let S' be a strategy for A different from S. Replace the first function S_1 of S with S_1', leaving S otherwise the same. By construction of S_1, this change cannot decrease the payoff. Now also replace the second function S_3 of S with S_3'; similarly this cannot decrease the payoff. Continuing in this way shows that S' does not do better than S for A. A similar argument works for T and B. ∎

Definition 6 *An* improvement function *I_A for player A is a tuple $(I_1, I_3, \ldots)$ of polynomial time machines. Each I_i takes a parameter a and inputs $x_1, \ldots, x_{i-1}$, can query oracles S and T for strategies, and outputs a move $x_k < a$. Clearly, given a, S and T, an improvement function defines a strategy $I_A(a, S, T)$ for A (when writing this strategy we will usually omit the parameter a). An improvement function I_B for B is defined similarly.*

Definition 7 *Given a parameter a and improvement functions I_A and I_B, an equilibrium with respect to I_A, I_B is a pair (S, T) of strategies satisfying*

$$v[I_A(S,T),T] \geq v[S,T] \wedge v[S, I_B(S,T)] \leq v[S,T]. \tag{18}$$

This expresses the idea that neither A nor B can unilaterally improve his strategy in polynomial time, even given knowledge of the other player's strategy.

Evaluating (18) is polynomial-time in a, S, T. In particular if $p(|a|)$ is a bound on the running time of the machines making up I_A and I_B, then evaluating (18) uses at most $2kp(|a|) + 2k$ queries to each of S and T. This means that to find an equilibrium, we do not need to find *total* strategies S and T, defined on all possible game histories. It is enough to find *partial* strategies, as long as they are defined on all queries made in (18), and satisfy it; this is because we could extend them arbitrarily to total strategies, and they would still satisfy it. Since this is only a polynomial number of queries, we can code such partial strategies as numbers less than $a^{2kp(|a|)+2k}$ (the number of possible sequences of oracle replies). Quantifying over them thus collapses to normal "first-order" bounded quantification. This allows us to turn the principle that an equilibrium exists into a search problem in **TFNP**.

Definition 8 *A $\mathbf{PE}_k$ search problem (standing for* polynomial time equilibrium*) is defined by a payoff function v and improvement functions I_A and I_B. The problem is: given a parameter a, find a pair of partial strategies S and T which are in equilibrium with respect to I_A and I_B.*

Theorem 7.2 *The class $\mathbf{PE}_k$ of search problems characterizes the $\forall\hat{\Sigma}^b_1$ sentences provable in T^k_2, by reductions formalizable in T^0_2.*

Proof. One direction is immediate: T^k_2 proves everything we need about alternating minima and maxima of length k, and in fact is strong enough to formalize polynomial time functions with Σ^p_k oracles. So in a model of T^k_2 we can simulate a computation of (18) using the true equilibrium min-maxing strategies of Theorem 7.1. We store every oracle query and reply made to S and T in this computation, and these lists of queries and replies give us our partial strategies.

For the other direction, we will give a reduction of $\mathbf{GPLS}_k$ to $\mathbf{PE}_k$. Suppose that an instance of $\mathbf{GPLS}_k$ is given by functions $v, h_1, \ldots, h_k$. Recall that the problem is, given a parameter a (which every function takes as a first argument, but which we will leave unwritten for clarity), to find $x_1, x_3, \ldots$ and $y_2, y_4, \ldots$ such that

$$v(x_1, h_2(x_1, y_2), x_3, h_4(x_1, y_2, x_3, y_4), \ldots) \leq v(h_1(x_1), y_2, h_3(x_1, y_2, x_3), y_4, \ldots).$$

We define an instance of $\mathbf{PE}_k$. The payoff function will be exactly v (again we will not write the first argument a). The improvement function $I_A = (I_1, I_3, \ldots)$ for A will only query A's strategy $S = (S_1, S_3, \ldots)$ and will not use B's strategy. The idea is that for each (odd) j the function I_j is, roughly speaking, the composition $h_j \circ S_j \circ h_{j-1}$. More precisely, $I_j(y_2, y_4, \ldots, y_{j-1})$ is calculated as follows, in $j + 1$ steps:

- At step 1, set $x_1 = S_1()$;
- At step 2, set $x_2 = h_2(x_1, y_2)$;
- Then at odd steps $i = 3, \ldots, j$, set $x_i = S_i(x_2, x_4, \ldots, x_{i-1})$;
- And at even steps $i = 2, \ldots, j-1$, set $x_i = h_i(x_1, y_2, \ldots, y_i)$;
- Finally at step $j+1$ output $h_j(x_1, y_2, \ldots, y_{j-1}, x_j)$.

Similarly the idea for the improvement function $I_B = (I_2, I_4, \ldots)$ for B is that for each (even) j, the function I_j is the composition $h_j \circ T_j \circ h_{j-1}$. Precisely, $I_j(x_1, x_3, \ldots, x_{j-1})$ is calculated as follows, in $j+1$ steps:

- At odd steps $i = 1, \ldots, j-1$, set $y_i = h_i(x_1, y_2, \ldots, x_i)$;
- At even steps $i = 2, \ldots, j$, set $y_i = T_i(y_1, y_3, \ldots, y_{i-1})$;
- Finally at step $j+1$ output $h_j(x_1, y_2, \ldots, x_{j-1}, y_j)$.

Now suppose (S, T) is an equilibrium for I_A and I_B. Let $\bar{y}$ be a play of I_A against T, and let $\bar{x}$ be the internal values used by I_A as described in the definition of I_A above. Then we have the following (in item 2, y_j is the output of $I_j(y_2, \ldots, y_{j-1})$):

1. For each odd j, $x_j = S_i(x_2, x_4, \ldots, x_{j-1})$;
2. For each odd j, $y_j = h_j(x_1, y_2, \ldots, y_{j-1}, x_j)$;
3. For each even j, $y_j = T_j(y_1, y_3, \ldots, y_{j-1})$;
4. For each even j, $x_j = h_j(x_1, y_2, \ldots, x_{j-1}, y_j)$.

Since S and T are in equilibrium, $v(\bar{y}) = v[I_A, T] \geq v[S, T]$. On the other hand, if we let $\bar{x}$ be a play of S against I_B and let $\bar{y}$ be the internal values used by I_B, then $\bar{x}$ and $\bar{y}$ will have exactly the same values as above, and by equilibrium we get that $v(\bar{x}) = v[S, I_B] \leq v[S, T]$. Thus we have sequences $\bar{x}$ and $\bar{y}$ such that $v(\bar{x}) \leq v(\bar{y})$ and where each even $x_j = h_j(x_1, y_2, \ldots, y_j)$ and each odd $y_j = h_j(x_1, y_2, \ldots, x_j)$, exactly as required for a solution of our instance of $\mathbf{GPLS}_k$. ■

This theorem relativizes (as does Theorem 6.1). Hence to prove a relativized separation of the $\forall\hat{\Sigma}_1^b$ consequences of T_2^{k+1} from those of T_2^k it is sufficient to find an oracle with respect to which finding a feasible equilibrium is strictly harder for $(k+1)$-turn games than it is for k-turn games.

From the proof we can also draw the corollary that the general problem of finding a feasible equilibrium is always reducible to an instance where the improvement functions have the rather simple form that arises from $\mathbf{GPLS}_k$.

References

[1] C. Àlvarez, J. Gabarró and M. Serna. Polynomial Space Suffices for Deciding Nash Equilibria Properties for Extensive Games with Large Trees, *LNCS 3827, Proceedings of ISAAC*, 2005, pp. 634-643.

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

[3] A. Beckmann and S.R. Buss. Polynomial Local Search in the Polynomial Hierarchy and witnessing fragments of Bounded Arithmetic, *Swansea Univ. Research Report CSR 15-2008*.

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

[5] 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.

[6] X. Chen and X. Deng. Settling the Complexity of 2-Player Nash-Equilibrium, *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science* 2006, pp. 261 - 272.

[7] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium, *Communications of the ACM 52(2),* 2009, pp. 89-97.

[8] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pp. 604-612.

[9] 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), 1995, pp. 79-88.

[10] E. Jeřábek. The strength of sharply bounded induction, *Mathematical Logic Quarterly* 52(6), 2006, pp. 613-624.

[11] D.S. Johnson, C.H. Papadimitriou and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences* 37, 1988, pp. 79-100.

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

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

[14] O. Morgenstern and J. von Neumann, *The Theory of Games and Economic Behaviour,* Princeton University Press, 1947.

[15] P. Pudlák, Fragments of Bounded Arithmetic and the lengths of proofs, *Journal of Symbolic Logic*, 73(4), 2008, pp. 1389-1406.

[16] P. Pudlák, On search problems and $\forall\Sigma^b_1$ theorems of bounded arithmetic. Talk at the *Prague-Vienna workshop on Proof Theory and Proof Complexity,* 2006.

[17] A. Skelley and N. Thapen. The provably total search problems of bounded arithmetic, preprint, 2007.