Lifting independence results in bounded arithmetic

Mario Chiari* Jan Krajíček†

Mathematical Institute, Academy of Sciences, Prague

Abstract

We investigate the problem how to lift the non - $\forall\Sigma_1^b(\alpha)$ - conservativity of $T_2^2(\alpha)$ over $S_2^2(\alpha)$ to the expected non - $\forall\Sigma_i^b(\alpha)$ - conservativity of $T_2^{i+1}(\alpha)$ over $S_2^{i+1}(\alpha)$, for $i > 1$. We give a non-trivial refinement of the "lifting method" developed in [4, 8], and we prove a sufficient condition on a $\forall\Sigma_1^b(f)$-consequence of $T_2(f)$ to yield the non-conservation result.

Further we prove that Ramsey's theorem, a $\forall\Sigma_1^b(\alpha)$ - formula, is not provable in $T_2^1(\alpha)$, and that $\forall\Sigma_j^b(\alpha)$ - conservativity of $T_2^{i+1}(\alpha)$ over $T_2^i(\alpha)$ implies $\forall\Sigma_j^b(\alpha)$ - conservativity of the whole $T_2(\alpha)$ over $T_2^i(\alpha)$, for any $j \geq 2$.

Introduction

Bounded arithmetic theory $T_2(\alpha)$ is a first-order theory of arithmetic in the language $L(\alpha)$ that contains function symbols:

$$0,\ 1,\ x+y,\ x\cdot y,\ \lfloor\frac{x}{2}\rfloor,\ |x|,\ x\#y$$

where $|x|$ is the length of the binary representation of x and $x\#y$ is $2^{|x||y|}$, and predicate symbols:

$$x = y,\ x \leq y,\ \alpha(x_1,\ldots,x_k)\ ,$$

where $\alpha(x_1,\ldots,x_k)$ is a k-ary predicate symbol without any particular interpretation in the standard model (k may vary). $T_2(\alpha)$ is axiomatized by a finite set $BASIC$ of bounded first order axioms that contains no axioms about α and by the induction scheme for all bounded formulas in the language $L(\alpha)$.

*1991 Mathematics Subject Classification. Primary 03F30; Secondary 68Q15. Supported by a grant of the Education Ministry of the Czech Republic.

†Partially supported by a grant jointly supported by the NSF and the Education Ministry of the Czech Republic, and by the grant #$A1019602$ of the Czech Academy of Sciences.
Email: krajicek@math.cas.cz

The class of bounded formulas of $L(\alpha)$ is stratified into levels $\Sigma_0^b(\alpha) \subset \Sigma_1^b(\alpha) \subset \ldots$ analogously to the arithmetical hierarchy $\Sigma_0^0 \subset \Sigma_1^0 \subset \ldots$. In particular, $\Sigma_i^b(\alpha)$ is the class of bounded formulas with $(i-1)$ alternations of bounded quantifiers, starting with an existential one and not counting the sharply bounded ones (i.e., quantifiers bounded by the a term of the form $|t|$). $\Sigma_i^b(\alpha)$-formulas define exactly the predicates in the i^{th} level $\Sigma_i^p(\alpha)$ of the polynomial time hierarchy relative to the oracle α.

$T_2^i(\alpha)$ is obtained from $T_2(\alpha)$ by restricting the induction scheme to $\Sigma_i^b(\alpha)$-formulas only. Another important family of subtheories of $T_2(\alpha)$ are $S_2^i(\alpha)$ that are based on the polynomial - induction scheme for $\Sigma_i^b(\alpha)$-formulas $A(a)$:

$$[A(0) \wedge \forall x; A(\lfloor \frac{x}{2} \rfloor) \rightarrow A(x)] \ \rightarrow \ \forall x; A(x)$$

The following is known about the conservativity relations between the theories $T_2^i(\alpha)$ and $S_2^i(\alpha)$:

1. $S_2^i(\alpha) \subseteq T_2^i(\alpha) \subseteq S_2^{i+1}(\alpha)$ (in [2])
2. $S_2^{i+1}(\alpha)$ is $\forall\Sigma_{i+1}^b(\alpha)$-conservative over $T_2^i(\alpha)$ (in [3])
3. $S_2^{i+1}(\alpha) \neq T_2^i(\alpha)$ (in [10])

As $S_2^{i+1}(\alpha)$ is $\forall\Sigma_{i+2}^b(\alpha)$ - axiomatizable, all possible conservation relations between $S_2^{i+1}(\alpha)$ and $T_2^i(\alpha)$ are therefore decided. Furthermore, it holds that $T_2^i(\alpha) \neq S_2^i(\alpha)$ ([8]) and, in fact, that $T_2^i(\alpha)$ is not $\forall\Sigma_i^b(\alpha)$-conservative over $S_2^i(\alpha)$ ([4]).

In this paper we are interested in the question to prove non - $\forall\Sigma_{i-j}^b(\alpha)$ - conservativeness of $T_2^i(\alpha)$ over $S_2^i(\alpha)$, for some $j \geq 1$ and all $i \geq 2$. A step in this direction was made in [5] where it was shown that $T_2^2(\alpha)$ is not $\forall\Sigma_1^b(\alpha)$ - conservative over $S_2^2(\alpha)$. In the present paper we give a new method designed to lift the result of [5] to a non-$\forall\Sigma_i^b(\alpha)$-conservativity of $T_2^{i+1}(\alpha)$ over $S_2^{i+1}(\alpha)$, for all $i \geq 1$. The method (Theorem 2.1), based on Boolean complexity, refines the methods of [4, 8]. We obtain a sufficient condition on a formula such that if there is a $\forall\Sigma_1^b(f)$-consequence of $T_2(f)$ satisfying the condition then $T_2^{i+1}(\alpha)$ is not $\forall\Sigma_i^b(\alpha)$-conservative over $S_2^{i+1}(\alpha)$, for all $i > 1$ (Corollary 6.1). Although we conjecture the existence of such a formula we have not found it.

Further we give a new example of a $\forall\Sigma_1^b(\alpha)$ - consequence of $T_2(\alpha)$ not provable in $T_2^1(\alpha)$. In particular, we show that Ramsey's theorem is not provable in $T_2^1(\alpha)$ (Theorem 4.2). We also prove that possible conservativity relations among subsystems of $T_2(\alpha)$ contain no gaps, meaning that $\forall\Sigma_j^b(\alpha)$ - conservativity of $T_2^{i+1}(\alpha)$ over $T_2^i(\alpha)$ implies $\forall\Sigma_j^b(\alpha)$ - conservativity of the whole $T_2(\alpha)$ over $T_2^i(\alpha)$, for any $j \geq 2$ (Theorem 5.3).

The paper is organized as follows. First section recalls the PLS - witnessing theorem from [4]. The lifting method is motivated and formulated in the second section and proved in the third section. The independence of Ramsey's theorem is proved in the fourth section, the no-gaps theorem about conservativity among subsystems of $T_2(\alpha)$ is proved in the fifth section. A corollary for the non-conservativeness problem is derived in the last section.

Although we recall some notions and results from bounded arithmetic, we advice the reader to consult either the original papers or [2, 9] for details.

1 Witnessing via PLS - problems

In this section we briefly recall the witnessing theorem for $T^1_2(\alpha)$ characterizing provably total $\Sigma^b_1(\alpha)$-functions in terms of *polynomial local search* problem (PLS - problem) of [11]. An *instance* of a PLS-problem L is a number x (coding a finite string of $0, 1$ of its bits). An instance x determines a *solution* set $F_L(x)$:

$$F_L(x) = \{y \mid |y| \leq |x|^\ell\},$$

(ℓ is a constant depending on L). The *cost* function $C_L(x, s)$ assigns to a solution $s \in F_L(x)$ its cost. The *neighbourhood function* $N_L(x, s)$ is defined on $F_L(x)$ and for $s \in F_L(x)$ it holds that:

$$(*_L) \qquad N_L(x, s) = s \vee [\ N_L(x, s) \in F_L(x) \wedge C_L(x, N_L(x, s)) < C_L(x, s)\ ]\ .$$

If $N_L(x, s) \neq s$, $N_L(x, s)$ is a solution of a cost smaller than the cost s. It is required that the functions $C_L(x, s)$ and $N_L(x, s)$ are polynomial time computable.

The search task of the PLS-problem L is: given instance x, find a solution $s \in F_L(x)$ for which $N_L(x, s) = s$ (a *locally optimal* solution).

An oracle PLS-problem is defined similarly, with functions $C_L(x, s)$ and $N_L(x, s)$ being computed by polynomial-time oracle machines. An oracle PLS-problem L whose machines have access to a particular oracle α is called a PLS^α-problem and it is denoted $L(\alpha)$.

Theorem 1.1 ([4]) *Let $\Theta(x, y, \alpha)$ be a $\Sigma^b_i(\alpha)$-formula, $i \geq 1$. Assume that:*

$$T^i_2(\alpha) \vdash \forall x \exists y \Theta(x, y, \alpha)\ .$$

Then there is an oracle PLS-problem L and a $\Sigma^b_{i-1}(\alpha)$-formula $\beta(x, \alpha)$ defining for all $\alpha \subseteq \omega$ the set $\beta^\alpha = \{x \in \omega \mid \beta(x, \alpha)\}$ such that for all α:

1. *the condition $(*_L)$ from the definition of PLS-problems holds for the problem $L = L(\beta^\alpha)$*

2. *whenever $s \in F_{L(\beta^\alpha)}(x, s)$ is a locally optimal solution, then*

(a) s has the form $(y, z_1, \ldots, z_k)$

(b) $\Theta(x, y, \alpha)$ is valid.

In particular, the $\forall\exists\Sigma_1^b(\alpha)$-consequences of $T_2^1(\alpha)$ are witnessed by projections of PLS^α-problems.

2 The lifting method - a formulation

In this section we explain the method designed to "lift" a suitable separation between $S_2^2(\alpha)$ and $T_2^2(\alpha)$ to the theories $T_2^{i+1}(\alpha)$ and $S_2^{i+1}(\alpha)$, for all $i \geq 1$. A version of the lifting method that could suffice was conjectured in [4]. We develop another version of the method. The version of [8] was based on $P^{NP}[O(\log n)]$ - witnessing for $S_2^1(\alpha)$ while [4] used P^{NP} - witnessing for $S_2^2(\alpha)$. We shall use Theorem 1.1. These differences influence the combinatorial part of the methods but the general scheme is the same. We give all necessary details for the new material but the general structure of the method is only outlined and the reader should consult [4, 8, 9], and [7] if necessary, for details.

Let $\Phi(a, f)$ be a bounded formula with a unary function symbol $f(x)$. This is a perfectly general situation as any finite number of predicates and functions can be encoded by a unary function in a simple way. Let the formula $\Phi^*(a, \beta)$ be the translation of Φ obtained by replacing f by its bit-graph $\beta(x, j) := (f(x))_j = 1$, where $(f(x))_j$ is the j^{th} bit of $f(x)$. Namely, $f(x) = y$ can be written as the $\Delta_1^b(\beta)$-formula:

$$\forall j < |a|; \beta(x, j) \equiv ((y)_j = 1) \ .$$

and so any $\Sigma_i^b(f)$-formula $\Phi(f)$ translates in this way into a $\Sigma_i^b(\beta)$-formula $\Phi^*(\beta)$ (see also [4, 5]).

Let $A(x, j)$ be a $\Pi_i^b(\alpha)$-formula. Take the formula

$$\Psi(a, \alpha) \ := \ \Phi^*(a, \ \beta/A)$$

obtained from Φ^* by replacing all occurrences of β by A. Whenever there is an oracle PLS-problem L such that, for all f, $L(f)$ witnesses $\Phi(a, f)$ then there is an oracle PLS-problem L' such that $L'(\gamma)$ witnesses $\Psi(a, \alpha)$, where γ is a $\Sigma_i^p(\alpha)$-predicate. The idea of the lifting method is that if the formula A is sufficiently *generic* then, in a sense, the opposite implication holds too. That is, if Ψ is witnessed by some $PLS^{\Sigma_i^p(\alpha)}$-problem then Φ is *essentially* witnessed by a PLS^f-problem. The qualification *essentially* means that the witnessing will work for many, but not all, functions f. The generic formulas $A := A_i(x, j)$ have the form:

$$\forall z_1 < a \exists z_2 < a \ldots Q z_i < a, \ \alpha(x, j, z_1, \ldots, z_i) \ ,$$

with i alternating quantifiers.

In order to state the technical theorem, we need a few simple notions from Boolean complexity. We just sketch the basic idea. For formal definitions see [4].

We may think of a predicate $\alpha(x, j, \overline{z})$ as of a propositional variable $p_{x,j,\overline{z}}$ which is true iff $\alpha(x, j, \overline{z})$ holds. A propositional formula is Σ_t^1 iff it has a disjunctive normal form (DNF) in which every disjunct is a conjunction of at most t literals. A formula is Π_t^1 iff its negation is Σ_t^1, and it is Δ_t^1 iff it is equivalent to both Σ_t^1 and Π_t^1.

Consider a predicate $P(u)$ that is computable by a polynomial-time machine M^α with oracle α. For any fixed u, each particular accepting computation of M^α on the input u is determined by the oracle answers which, in turn, are determined by conjunctions of atoms $\alpha(x, j, \overline{z})$ (for affirmatively answered queries $[\alpha(x, j, \overline{z})?]$) and their negations (for negatively answered queries). These conjunctions have size at most t, if t is the time-bound of the machine on inputs of size $|u|$ (i.e., t does not depend on α). Hence, $P(u)$ is true (i.e., M^α accepts u) iff the disjunction of the conjunctions determined by all possible accepting computations is valid. Hence, $P(u)$ as well as $\neg P(u)$ (taking the rejecting computations in place of the accepting ones) are expressible by a Σ_t^1-formula. Thus, $P(u)$ is, in fact, expressible by a Δ_t^1-formula.

A predicate $P(u)$ that is expressible by Δ_t^1-formulas might not be obtained in this way from a predicate that is computable by an oracle polynomial-time machine, as the sequence of formulas might not be even recursive. All of them, however, share an important property: to fix the truth value of $P(u)$ for a particular u, one needs to specify at most t truth values of the predicate $\alpha(x, j, \overline{z})$, as to force the truth of a conjunction in the DNF of either $P(u)$ or $\neg P(u)$. Typically, t is small.

Now we state a technical theorem, our refinement of the lifting method.

Theorem 2.1 *Let $i \geq 1$. There is a translation $^{(i)}$ of formulas to formulas:*

$$\Phi(a, f) \mapsto \Phi^{(i)}(a, \alpha)$$

such that, for all $k \geq 1$, $\Phi^{(i)} \in \Sigma_{k+i}^b(\alpha)$ if $\Phi \in \Sigma_k^b(f)$, and which has the following two properties.

1. *For all $j \geq 1$, if $T_2^j(f) \vdash \Phi$ then $T_2^{j+i}(\alpha) \vdash \Phi^{(i)}$.*
2. *If there is an oracle PLS-problem L' and a predicate γ such that:*
   (a) *the predicate γ is $\Sigma_i^b(\alpha)$-definable,*
   (b) *for all α and all a, the problem $L'(\gamma)$ witnesses the formula $\Phi^{(i)}(a, \alpha)$ in the sense of Theorem 1.1,*

   then there is a constant p such that for any sufficiently large $a := m$ there is an oracle PLS-problem L, a subset $D \subseteq m$, a number s and an oracle η such that:

(a) $s \leq (\log m)^p$,

(b) η *is definable by* Δ^1_s *- formulas that are formed from the atoms* $(f(x))_j = 1$,

(c) $|D \cap [w \cdot m^{3/4}, (w+1) \cdot m^{3/4})| \geq m^{1/8}$, *for all* $w < m^{1/4}$

(d) *the* PLS^η *- problem* $L(\eta)$ *witnesses the formula* $\Phi(m, f)$ *for all functions* $f : m \to m$, *such that* $f(x) = 0$ *if* $x \notin D$ *and* $f(x) \in [m - m^{1/4}, m)$ *if* $x \in D$.

Next section is devoted to the proof of the theorem. The proof is summarized after the proof of Lemma 3.3.

3 The lifting method - a construction

We shall define the translation $\Phi^{(i)}$ to be simply the formula $\Psi(a, \alpha)$ defined at the beginning of the previous section with $A := A_i$. There are difficulties encountered when trying to prove the theorem by the methods of [4, 8]. We will first examine these difficulties to motivate the modification of the methods needed for the proof of the theorem.

There are $2^{(\log m)^k}$ possible oracle queries $[\gamma(u)?]$, $u < 2^{(\log m)^k}$, where k is a constant depending on the PLS-problem L' only. Because the predicate γ is $\Sigma^b_i(\alpha)$-definable, each query $[\gamma(u)?]$ is expressible by a depth $i+1$ circuit of size $2^{(\log m)^{O(1)}}$, with the top-most connective $\bigvee$, with alternating levels of $\bigwedge$ and $\bigvee$, and such that the level 1 connectives have arity at most $(\log m)^{O(1)}$.

At the same time formulas $(f(x))_j = 1$ in Φ^* are expressed by the formulas $A_i(x, j)$ which are depth i circuits of size $m^{O(1)}$ with the top-most connective $\bigwedge$, alternating levels of $\bigwedge$ and $\bigvee$, and with each connective of arity m.

[7] showed that, for every $i \geq 1$, there is a partial truth assignment $\tilde{\rho}$ with two properties:

1. each $\gamma^{\tilde{\rho}}(u)$ is expressible by a depth i, size $2^{(\log m)^{O(1)}}$ circuit with the bottom connectives of the arity $(\log m)^{O(1)}$, for all $u < 2^{(\log m)^k}$,

2. each $A^{\tilde{\rho}}_i(x, j)$ computes the formula $A_{i-1}(x, j)$.

($\gamma^{\tilde{\rho}}$ and $A^{\tilde{\rho}}_i$ denote the formulas with the atoms from the domain of $\tilde{\rho}$ evaluated by $\tilde{\rho}$). These are Lemma 4 in Section 5.3 and Lemma 8 in Section 5.2 of [7], with $\tilde{\rho} := \rho g(\rho)$ (see below).

By applying these lemmas $(i-1)$-times, we may reduce the general statement of the theorem for $i > 1$ to the case $i = 1$.

The case $i = 1$ requires a modification of the construction from [7]. To understand the problem with the original construction, we first analyze what happens in the case of the restrictions used in [7]. Queries $[\gamma(u)?]$ are expressed by Σ_t^1-circuits, $t = (\log m)^{O(1)}$, and statements $(f(x))_j = 1$ are expressed by conjunctions of the form:

$$\bigwedge_{z<m} \alpha(x,j,z) \ .$$

For $x < m$ and $j < \log m$ the atoms $\alpha(x,j,z)$ are partitioned into blocks:

$$B_{x,j} := \{\alpha(x,j,z) \mid z < m\} \ .$$

Fix $0 < \delta < 1$ and $q := m^{\delta-1}$. For each block we choose independently a value $s_{x,j}$:

$$s_{x,j} := \begin{cases} * & \text{with probability } q \\ 0 & \text{with probability } 1-q. \end{cases}$$

Then we assign independently a value $\rho(\alpha(x,j,z))$:

$$\rho(\alpha(x,j,z)) := \begin{cases} s_{x,j} & \text{with probability } q \\ 1 & \text{with probability } 1-q. \end{cases}$$

Restrictions ρ are not sufficient, however, for the argument in [7] and an additional set of restrictions is introduced there. For those blocks $B_{x,j}$ for which there are some $\alpha(x,j,z) \in B_{x,j}$ receiving the value * by ρ, we define:

$$g(\rho)(\alpha(x,j,z)) := 1$$

for all $\alpha(x,j,z)$ which received the value * by ρ except for one, say the one with the minimal z. In other words, $g(\rho)$ changes in each block all but one * into 1.

For any fixed $x < m$, the expected number of $s_{x,j}$, $j < \log m$, receiving * is $q \cdot \log m$, which goes to 0. Hence, with high probability, for most $x < m$ (about $m - m^{\delta}$) all $s_{x,j}$ are 0 and for the other there is exactly one $j \leq \log m$ such that $s_{x,j} = *$.

If $s_{x,j} = 0$ then the expected number of $\alpha(x,j,z)$ receiving 0 by ρ is $q \cdot m = m^{\delta} >> 1$. Hence, with a high probability, for those x, j, each conjunction $\beta(x,j) = \bigwedge_z \alpha(x,j,z)$ is collapsed by $\tilde{\rho}$ to 0. Together with the remark above this means that after the restriction $\tilde{\rho}$, with high probability, $f(x) = 0$ for most $x < m$ and $f(x)$ has only one bit undefined for the other $x < m$. Thus, witnessing $\Phi(m, f)$ for functions f respecting the values determined by $\tilde{\rho}$ cannot be expected to be hard.

The first problem that, for most x, all $s_{x,j}$ are 0 is overcomed in the following way.

Define a new partition of these atoms into new blocks:

$$B_x := \{\alpha(x,j,z) \mid j \leq \log m, z < m\} \ .$$

For an interval $[a,b) \subseteq m$ of length $m^{1-\frac{\delta}{2}}$ the expected number of blocks B_x, $x \in [a,b)$, receiving value $s_x = *$ is $q \cdot m^{1-\frac{\delta}{2}} = m^{\delta/2}$. Hence, with a high probability, each interval $[w \cdot m^{1-\frac{\delta}{2}}, (w+1) \cdot m^{1-\frac{\delta}{2}})$, $w = 0, \ldots, m^{\frac{\delta}{2}}-1$, contains at least $m^{\delta/4}$ elements x such that $s_x = *$.

Even for $s_x = *$ all but one atoms in B_x receive 1 by $\rho g(\rho)$ and hence leave at most two possible values for $f(x)$. This is caused solely by $g(\rho)$ because for any fixed x and j the expected number of $\alpha(x,j,z)$ receiving $*$ by ρ is $q \cdot m = m^\delta >> 1$, and hence ρ alone does not set any bit of $f(x)$. This suggests that instead of leaving one $*$ in each block we leave more of them.

Definition 3.1 *Let $0 < \delta, \epsilon < 1$ be two constants. Let m be fixed, and let the atoms $\alpha(x,j,z)$ and the blocks B_x be as above.*

We define the probability space $R_m(\delta,\epsilon)$ of random restrictions $\tilde{\rho}$ defined by the following process.

Put $q := m^{\delta-1}$ and define independently for all $x < m$:

$$s_x := \begin{cases} * & \text{with the probability } q \\ 0 & \text{with the probability } 1-q, \end{cases}$$

and then for $\alpha(x,j,z) \in B_x$:

$$\rho(\alpha(x,j,z)) := \begin{cases} s_x & \text{with the probability } q \\ 1 & \text{with the probability } 1-q \end{cases}$$

For any ρ, define $h(\rho)$ as follows:

(a) pick the first $\epsilon \cdot \log m$ numbers $j_s \leq \log m$, $s = 0, \ldots, \epsilon \log m - 1$, for which some $\alpha(x,j_s,z)$ receives the value $*$,

(b) for each such j_s take the smallest $z_s < m$ such that $\rho(\alpha(x,j_s,z_s)) = *$,

(c) change all those $\alpha(x,j,z)$ which received $*$ into 1 except those $\epsilon \cdot \log m$ atoms $\alpha(x,j_s,z_s)$.

Let $\tilde{\rho}$ be the composition $\rho h(\rho)$.

Note that with high probability j_s are just the first s bits, i.e., $j_s = s$ (as for fixed x and j the expected number of $\alpha(x,j,z)$ receiving $*$ is $q \cdot m = m^\delta >> 1$). The proof of the following lemma is just a formal verification of the explanation above: see [7], Section 5.2, Lemma 8.

Lemma 3.2 *Let $0 < \delta, \epsilon < 1$ and let $\tilde{\rho} = \rho h(\rho)$ be chosen at random from $R_m(\delta,\epsilon)$. Then, for sufficiently large m, with probability at least $\frac{1}{4}$ the following holds for all $w < m^{\frac{\delta}{2}}$:*

$$\forall x \in [w \cdot m^{1-\frac{\delta}{2}}, (w+1) \cdot m^{1-\frac{\delta}{2}}); (\forall s < \log m \exists z < m; \tilde{\rho}(\alpha(x,s,z)) = 0) \vee$$

$$\vee\ (\forall s < \log m \forall z < m; \tilde{\rho}(\alpha(x,s,z)) = * \vee \tilde{\rho}(\alpha(x,s,z)) = 1)$$

and

$$\exists D_w \subseteq [w \cdot m^{1-\frac{\delta}{2}}, (w+1) \cdot m^{1-\frac{\delta}{2}});\ |D_w| \geq m^{\delta/4}\ \wedge$$
$$\forall x \in D_w\ \forall s < \log m;$$
$$(s < \epsilon \cdot \log m\ \rightarrow\ \exists z < m;\ \tilde{\rho}(\alpha(x,s,z)) = *)\ \wedge$$
$$\wedge\ (\log m > s \geq \epsilon \cdot \log m\ \rightarrow\ \forall z < m;\ \tilde{\rho}(\alpha(x,s,z)) = 1)$$

In order to switch all $2^{(\log m)^k}$ Σ^1_t-formulas expressing the possible oracle queries into Δ^1_s-formulas, for some $s = (\log m)^{O(1)}$, we have to establish a variant of the switching lemma [7, Section 5.3, Lemma 4].

Lemma 3.3 *Let $0 < \delta, \epsilon < 1$ and assume that $\delta + \epsilon < 1$. Then there is a constant $0 < \xi < 1$ such that, for every sufficiently large m, for every $u < 2^{(\log m)^k}$, and for random $\tilde{\rho} \in R_m(\delta, \epsilon)$, the probability that $(\gamma(u))^{\tilde{\rho}}$ cannot be expressed as a Δ^1_s-formula is at most:*

$$\xi^s\ .$$

Proof :

The proof parallels the proof of Lemma 4 through Lemma 5 in Section 5.3 of [7] (the constant α there is denoted ξ here).

The first place where the particular form of $g(\rho)$ plays a role in that proof is the final computation of the probability after Lemma 7 on p. 165 of [7]. The fact that $g(\rho)$ leaves at most one $*$ in each block is used when the number of variables with the value $*$ and that are outside the set Y of the exposed blocks is expresses by the term $s - |Y|$.

In our case this expression is $s - \epsilon \cdot \ell \cdot |Y|$, where $\ell = \log m$. The bound to the number of all evaluations of all atoms outside the exposed blocks Y with the value $*$ is

$$2^{\epsilon \cdot \ell \cdot |Y|} \cdot \xi^{s - \epsilon \cdot \ell \cdot |Y|}\ .$$

Thus, the estimate of the sum in the final part of the proof of Lemma 5 in [7] changes into:

$$\sum_{Y \subset [r], Y \neq \emptyset} (2q)^{|Y|} 2^{\epsilon \cdot \ell \cdot |Y|} \xi^{s - \epsilon \cdot \ell \cdot |Y|}\ =\ \xi^s \sum_{i=1}^{r} \binom{r}{i} (\frac{2q 2^{\epsilon \cdot \ell}}{\xi^{\epsilon \cdot \ell}}\ )^i\ \leq$$

$$\leq \xi^s [(1 + \frac{2q2^{\epsilon \cdot \ell}}{\xi^{\epsilon \cdot \ell}}\ )^r\ - 1\ ]\ .$$

If:

(A)

$$\frac{2q2^{\epsilon\cdot\ell}}{\xi^{\epsilon\cdot\ell}} \;\leq\; 2^{\frac{1}{t}} - 1 \;,$$

then the sum is bounded by (as $r \leq t$ in [7]):

$$\xi^{s}(2^{\frac{r}{t}} - 1) \leq \xi^{s} \;.$$

The left side of the inequality (A) is equal to:

$$\frac{2m^{\delta-1}m^{\epsilon}}{m^{\epsilon\cdot\log\xi}} = 2m^{\delta+\epsilon-1-\epsilon\cdot\log\xi} \;.$$

Hence, it is sufficient to have:

(B)

$$2m^{\delta+\epsilon-1-\epsilon\cdot\log\xi} \leq \frac{2}{3t} \;,$$

as $\frac{2}{3t} \leq 2^{\frac{1}{t}} - 1$, for $t \geq 2$.

Because t is only $(\log m)^{O(1)}$, the inequality (B) is satisfied whenever:

$$\delta + \epsilon - 1 - \epsilon \cdot (\log\xi) < 0 \;;$$

i.e.,

(C)

$$\frac{\epsilon + \delta - 1}{\epsilon} < \log\xi \;.$$

By the hypothesis of the lemma, $\epsilon + \delta - 1 < 0$ hence $\frac{\epsilon+\delta-1}{\epsilon} < 0$ and we can find a constant $\xi < 1$ such that (C) holds.

This proves the lemma.

q.e.d.

By the last lemma, the probability that a random $\tilde{\rho} \in R_m(\delta, \epsilon)$ switches all queries $\gamma(u)$, $u < 2^{(\log m)^k}$, into Δ^1_s-formulas is at least:

$$\geq 1 - \xi^{s} \cdot 2^{(\log m)^k}$$

which is > 0 for $s = (\log m)^p$, p a sufficiently big constant.

Now we may go back to the proof of Theorem 2.1.

Proof of Theorem 2.1:

Define the formula $\Phi^{(i)}$ by:

$$\Phi^{(i)}(a,\alpha) := (\Phi(a,f))^*(\beta(x,j)/A_i(x,j))$$

replacing the bit graph β of f in Φ^* by the formula A_i

$$\forall z_1 < a \exists z_2 < a \dots Q z_i < a,\ \alpha(x,j,z_1,\dots,z_i)\ ,$$

Note that, if $\Phi \in \Sigma_k^b(f)$ then the formula $\Phi^{(i)}$ obtained in this way is $\Sigma_{k+i}^b(\alpha)$, as it is required. Also note that if some theory $T_2^j(f)$ proves Φ then $T_2^{j+i}(\alpha)$ proves $\Phi^{(i)}$; just translate a $T_2^j(f)$-proof of Φ into a $T_2^{j+i}(\alpha)$-proof of $\Phi^{(i)}$.

The proof of the second part of the theorem proceeds as follows:

1. Let L' be an oracle PLS-problem and let γ be a $\Sigma_i^b(\alpha)$-oracle satisfying the hypothesis of part 2 of the theorem.

2. Apply $(i-1)$-times the restrictions $\tilde{\rho} = \rho g(\rho)$ of [7], collapsing the queries to the oracle γ to Σ_t^1-formulas, $t = (\log m)^{O(1)}$, and the definitions of the bits of the values of the function f into conjunctions. The relevant computations are exactly those in [4], Theorem 2.6, steps 3-9.

3. Apply then a random restriction $\tilde{\rho} \in R_m(\delta,\epsilon)$ from Definition 3.1, for $\delta = \frac{1}{2}$ and $\epsilon = \frac{1}{4}$. By Lemma 3.3 it further switches the queries into Δ_s^1-circuits, for some $s = (\log m)^p$. Let η be the oracle defined by these circuits.

4. By Lemma 3.2, in each interval $[w \cdot m^{1-\frac{\delta}{2}}, (w+1)\cdot m^{1-\frac{\delta}{2}})$ there is at least $m^{\delta/4}$ elements x such that the first $\frac{1}{4}\cdot \log m$ bits of $f(x)$ are undetermined by $\tilde{\rho}$, and the rest of bits are 1. Let $D \subseteq m$ be the set of such x.

5. 4. means that for all $x \in D$, $f(x)$ can have any value in $[m - m^{\frac{1}{4}}, m)$. Hence the PLS^η-problem $L(\eta)$ witnesses the formula $\Phi(m,f)$ for all functions $f : m \to m$, such that $f(x) = 0$ if $x \notin D$ and $f(x) \in [m - m^{1/4}, m)$ if $x \in D$.

q.e.d.

4 Independence of Ramsey's theorem

We show in this section that a $\Sigma_1^b(\alpha)$ - formula expressing Ramsey's theorem

$$n \longrightarrow (\lfloor \frac{\log n}{2} \rfloor)_2^2$$

is not provable in $S_2^2(\alpha)$. Pudlák [12] proved that it is provable in $T_2(\alpha)$ by reducing it to the weak pigeonhole principle. In fact, it is provable in $T_2^5(\alpha)$ as computed in [9, Thm. 12.1.3].

Definition 4.1 *Let $\alpha(x,y)$ be a binary relation symbol. Formula $Ram(a,\alpha)$ is the formula:*

$$\exists X \subseteq \{0,\dots,a-1\};\ |X| = \lfloor \frac{|a|}{2} \rfloor \wedge$$
$$[(\forall x,y \in X; x \neq y \to \alpha(x,y)) \ \vee \ (\forall x,y \in X; x \neq y \to \neg\alpha(x,y))]$$

Note that $Ram(a,\alpha)$ is a $\Sigma_1^b(\alpha)$ - formula as X has sharply bounded cardinality and hence the subformulas starting with the quantifiers $\forall x,y \in X$ are equivalent (provably in S_2^1) to $\Delta_1^b(\alpha)$-formulas.

Theorem 4.2 *$S_2^2(\alpha)$ does not prove the formula $Ram(a,\alpha)$.*

Proof :

By the $\forall\Sigma_1^b$ - conservativity of $S_2^2(\alpha)$ over $T_2^1(\alpha)$ and by Theorem 1.1 it suffices to show that Ram is not witnessed by a PLS^α - problem. Assume on the contrary that $L(\alpha)$ is an oracle PLS - problem with a neighborhood function N_L and a cost function C_L. Let n^ℓ be a common time bound for some machines computing N_L and C_L. Pick a large enough (such that $a > 4(\log a)^\ell$) and let $n := |a|$.

By [6] there is a graph of size $2^{s/2}$ without a clique or an independent set of size s (see also [1] for further information). In particular, there is a graph of size $4n^\ell$ having no clique or independent set of size $\frac{n}{2}$. Fix one such graph G.

Call a computation of N_L and C_L *good* iff the oracle answers are consistent with a subgraph of G. Note that at most $2n^\ell$ oracle queries occur in any (two) computations of N_L and C_L and hence such a subgraph has size at most $4n^\ell$. As a is large enough, at least one good computation exists. Let c_0 be the minimal cost obtainable in a good computation, $c_0 = C_L^\alpha(a, s_0)$, where s_0 is the corresponding solution.

Let H be the subgraph of G according to which the oracle queries were answered in the computation of c_0. Solution s_0 is locally optimal in all $L(\alpha)$ such that $\alpha \supseteq H$. By the assumption s_0 must have the form

$$s_0 = (X, z_1, \dots, z_k)$$

such that X is either a clique of size $\frac{n}{2}$ in every graph (a,α), for $\alpha \supseteq H$, or an independent set of size $\frac{n}{2}$ in every such graph. However, that is impossible as $G \supseteq H$ has neither a clique or an independent set of the prescribed size.

q.e.d.

5 On gaps in conservativity relations

In this section we are going to show that in conservativity relations among subsystems of $T_2(\alpha)$ there are no gaps. In particular, if $T_2(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^i(\alpha)$ then already $T_2^{i+1}(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^i(\alpha)$, provided $j \geq 2$.

Lemma 5.1 *Let $k \geq 1$ and let $A(a, \alpha)$ be a bounded formula. Then the following two propositions are equivalent:*

1. $T_2^{k+1}(\alpha) \vdash A(a, \alpha)$
2. *There is a $\Sigma_1^b(\alpha)$ - formula $B(x, \alpha)$ and a term $t(a)$ such that* $T_2^k(\alpha, \beta) \vdash (\forall x \leq t(a);\ \beta(x) \equiv B(x, \alpha)) \ \rightarrow \ A(a, \alpha)$

Proof :

To prove that 2. implies 1. substitute in a $T_2^k(\alpha, \beta)$ - proof $B(x, \alpha)$ for $\beta(x)$. For the opposite implication encode by β the $\Sigma_1^b(\alpha)$ - parts of the induction formulas (in fact, we may assume that only one induction formula is used).

q.e.d.

Lemma 5.2 *Let $j \geq 2$, $k \geq i \geq 1$. Assume that $T_2^{k+1}(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^{i+1}(\alpha)$.*

Then $T_2^k(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^i(\alpha)$.

Proof :

Take a $\Sigma_j^b(\alpha)$ - formula $A(a, \alpha)$ that is provable in $T_2^{k+1}(\alpha)$ but not in $T_2^{i+1}(\alpha)$. Let $B(x, \alpha)$ and $t(a)$ be a formula and a term guaranteed by Lemma 5.1, and denote by $C(a, \alpha, \beta)$ the formula

$$(\forall x \leq t(a);\ \beta(x) \equiv B(x, \alpha)) \ \longrightarrow \ A(a, \alpha)$$

($C(a, \alpha, \beta)$ is $\Sigma_j^b(\alpha, \beta)$ - formula if $j \geq 2$). Then $T_2^k(\alpha, \beta)$ proves $C(a, \alpha, \beta)$ but $T_2^i(\alpha, \beta)$ does not (by Lemma 5.1).

Finally note, that two predicates α, β can be encoded in a quantifier - free way in one predicate.

q.e.d.

Theorem 5.3 *Let $j \geq 2$ and $i \geq 1$. Assume that $T_2(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $S_2^{i+1}(\alpha)$.*

Then $T_2^{i+1}(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $S_2^{i+1}(\alpha)$.

Proof :

From the hypothesis it follows that for some k, $T_2^k(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^i(\alpha)$.

Hence for some ℓ, $i \leq \ell < k$, $T_2^{\ell+1}(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^\ell(\alpha)$. Using Lemma 5.2 $(\ell - i)$ - times we get that $T_2^{i+1}(\alpha)$ is not $\forall\Sigma_j^b(\alpha)$ - conservative over $T_2^i(\alpha)$, and hence it is not $\forall\Sigma_j^b(\alpha)$ - conservative over $S_2^{i+1}(\alpha)$ either.

q.e.d.

6 Towards lifting the non - conservativity

In this section we formulate a sufficient condition on a $\forall\Sigma_1^b(f)$-formula such that the existence of a $\forall\Sigma_1^b(f)$-formula satisfying the condition implies a non-$\forall\Sigma_i^b(\alpha)$-conservativity of $T_2^{i+1}(\alpha)$ over $S_2^{i+1}(\alpha)$.

Corollary 6.1 *Assume that $\Phi(a,f)$ is a $\forall\Sigma_1^b(f)$-formula satisfying the following two conditions:*

1. $T_2(f) \vdash \Phi(a,f)$
2. *There is no constant p such that for any sufficiently large $a := m$ there is an oracle PLS-problem L, a subset $D \subseteq m$, a number s and an oracle η such that:*
   (a) $s \leq (\log m)^p$,
   (b) *η is definable by Δ_s^1 - formulas that are formed from the atoms $(f(x))_j = 1$,*
   (c) $|D \cap [w \cdot m^{3/4}, (w+1) \cdot m^{3/4})| \geq m^{1/8}$, *for all $w < m^{1/4}$*
   (d) *the PLS^η - problem $L(\eta)$ witnesses the formula $\Phi(m,f)$ for all functions $f : m \to m$, such that $f(x) = 0$ if $x \notin D$ and $f(x) \in [m - m^{1/4}, m)$ if $x \in D$.*

Then $T_2^{i+2}(\alpha)$ is not $\forall\Sigma_{i+1}^b(\alpha)$ conservative over $S_2^{i+2}(\alpha)$, for all $i \geq 1$.

Proof :
Take the $\forall\Sigma_{i+1}^b(\alpha)$-formula $\Phi^{(i)}(\alpha)$ provided by Theorem 2.1 from $\Phi(a,f)$. By the hypotheses of the theorem $T_2^\ell(f) \vdash \Phi(a,f)$, some $\ell \geq 2$, and by part 1 of Theorem 2.1 thus $T_2^{i+\ell}(\alpha) \vdash \Phi^{(i)}$. It suffices to show that $S_2^{i+2}(\alpha)$ does not prove $\Phi^{(i)}$, as by Theorem 5.3 then $T_2^{i+2}(\alpha)$ is not $\forall\Sigma_{i+1}^b(\alpha)$ - conservative over $S_2^{i+2}(\alpha)$.

Assume otherwise. By Theorem 1.1 there is an oracle PLS-problem L' and a predicate γ satisfying the hypotheses of part 2 of Theorem 2.1. Hence by Theorem 2.1 there is a constant p such that for sufficiently large $a := m$ there is L, D, s and η contradicting the hypotheses 2. of the theorem. That is a contradiction.

q.e.d.

Let us explain a construction of a $\forall\Sigma_1^b(f)$-formula satisfying the second condition of Corollary 6.1. Let $\Phi(a,g)$ be a $\forall\Sigma_1^b(g)$-formula such that $\Phi(m^{1/4},g)$ cannot be witnessed by an oracle PLS-problem $L(\eta)$ working with an oracle η computed by Δ_s^1-formulas, $s = (\log m)^{O(1)}$, from atoms $(g(x))_j = 1$.

Examples of such formulas $\Phi(a,g)$ are provided by (an encoding into one function g of) the formulas $WPHP$ or $Iter$ from [5], or by Ram from Theorem

4.2. The proofs of their unprovability in $S^2_2(g)$ go by showing that Φ cannot be witnessed by a PLS^g-problem. For $L(\eta)$, the same argument works almost literally. This is because the only property in those independence proofs that we used is:

> *every computation of $N_{L(g)}$ or of $C_{L(g)}$ is determined by only $(\log m)^{O(1)}$ values of g*

which holds also for an oracle η: one old query to g is now replaced by one query to η, whose truth-value is fixed by $(\log m)^{O(1)}$ values of g.

Now let f be a function $f : m \to m$ such that for every interval $I_w := [w \cdot m^{3/4}, (w+1) \cdot m^{3/4})$, $w < m^{1/4}$, there is exactly one value $g(w) \in [0, m^{1/4})$ such that:

1. $\forall x \in I_w, f(x) = 0 \vee f(x) = m - m^{1/4} + g(w)$
2. $\exists x \in I_w, f(x) \neq 0$.

Then $g(w)$ is $\Delta^b_1(f)$-definable:

$$g(w) = y \ \text{ iff } \ \exists x \in I_w, f(x) = m - m^{1/4} + y$$

and also

$$g(w) = y \ \text{ iff } \ \forall x \in I_w, f(x) \neq 0 \to f(x) = m - m^{1/4} + y$$

Clearly any function $g : m^{1/4} \to m^{1/4}$ may be encoded in this way by some $f : m \to m$ satisfying the conditions 1. and 2.. The $\Sigma^b_1(g)$-formula $\Phi(g)$ transforms into a $\Sigma^b_1(f)$-formula $\Phi^e(f)$ as g is $\Delta^b_1(f)$-definable. Moreover, $\Phi^e(m, f)$ cannot be witnessed by an oracle PLS-problem $L(\eta)$ satisfying the conditions of Corollary 6.1 as a witness to $\Phi^e(m, f)$ contains a witness to $\Phi(m^{1/4}, g)$ for the corresponding g.

It follows that the $\forall\Sigma^b_{i+1}(\alpha)$-formula $(\Phi^e)^{(i)}(\alpha)$ cannot be proved in $S^{i+2}_2(\alpha)$ as otherwise by Theorems 1.1 and 2.1 $\Phi^e(m, f)$ could be witnessed by an oracle PLS-problem L in a forbidden way.

The problem in applying Corollary 6.1 to such formulas $\Phi^e(f)$ is that to get $T_2(f) \vdash \Phi^e(f)$ (and hence $T^{i+\ell}_2(\alpha) \vdash (\Phi^e)^{(i)}(\alpha)$ for some ℓ by Theorem 2.1) one needs to assume the conditions 1. and 2. posed on f earlier (so that there is a function g for which $\Phi^e(m, f) \equiv \Phi(m^{1/4}, g)$). However, condition 2. is only $\Pi^b_2(f)$ rather than $\Pi^b_1(f)$.

Acknowledgement:

The first author wishes to thank the Czech Academy of Sciences for the generous hospitality and the Logic Group therein for making him feel very welcome in the beautiful city of Prague.

References

[1] Bollobás, B. (1985) *Random Graphs*, Academic Press.

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

[3] ——, (1990) Axiomatizations and conservation results for fragments of bounded arithmetic, in: *Logic and Computation*, Contemporary Mathematics, **106**: 57-84. Providence, American Mathematical Society.

[4] Buss, S. R., and Krajíček, J. (1994) An application of boolean complexity to separation problems in bounded arithmetic, *Proceedings of the London Mathematical Society*, **69(3)**: 1-27.

[5] Chiari, M., and Krajíček, J. (1995) Witnessing functions in bounded arithmetic and search problems, *Journal of Symbolic Logic*, to appear.

[6] Erdós, P. (1947) Some remarks on the theory of graphs, *Bulletin of the A.M.S.*, **53**: 292-294.

[7] Hastad, J. (1989) Almost optimal lower bounds for small depth circuits. in: *Randomness and Computation*, ed. S.Micali, Ser.Adv.Comp.Res. **5**: 143-170. JAI Press.

[8] Krajíček, J. (1993) Fragments of bounded arithmetic and bounded query classes, *Transactions of the A.M.S.*, **338(2)** : 587-598.

[9] —— (1995) *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications, Vol. **60**, *Cambridge University Press*, Cambridge - New York - Melbourne, 343 p.

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

[11] Papadimitriou, C. H., and Yannakakis, M. (1988) Optimization, approximation and complexity classes, in: *20th Annual ACM Symp. on Th. of Computing*, pp.229-234. ACM Press.

[12] Pudlák, P. (1992) Ramsey's theorem in bounded arithmetic, in :*Proc. Computer Science Logic*, Eds. Boerger E, et. al., Springer LN in Comp. Sci., **553**: 308-312.