A Theory for Log-Space and NLIN versus co-NLIN

Chris Pollett
214 MacQuarrie Hall
Department of Computer Science
San Jose State University
1 Washington Square, San Jose CA 95192
pollett@cs.sjsu.edu

August 10, 2003– Draft

Abstract

The use of Nepomnjaščiĭ's Theorem in the proofs of independence results for bounded arithmetic theories is investigated. Using this result and similar ideas, it is shown that at least one of S_1 or TLS does not prove the Matiyasevich-Robinson-Davis-Putnam Theorem. It is also established that TLS does not prove a statement that roughly means nondeterministic linear time is equal to co-nondeterministic linear time. Here S_1 is a conservative extension of the well-studied theory $I\Delta_0$ and TLS is a theory for LOGSPACE reasoning.

Mathematics Subject Classification: 03F30, 68Q15
Keywords: bounded arithmetic, independence results, MRDP

1 Introduction

In this paper applications of Nepomnjaščiĭ's Theorem to the provability of several important complexity statements in bounded arithmetic theories are considered. Recall that Nepomnjaščiĭ's Theorem states that those languages that can be decided in simultaneous time n^k, $k > 0$ and space n^ϵ, $1 > \epsilon$, the class $\mathsf{TISP}(n^k, n^\epsilon)$, are contained in the linear time hierarchy, LinH. The study of this theorem has recently undergone a renaissance since Fortnow [5] used it to prove time-space lower bounds for SAT.

The theory $I\Delta_0$ consists of defining axioms for the symbols of arithmetic together with induction for bounded formulas. By Wrathall [20] it is known that the Δ_0-predicates in this language are the predicates computable in the linear time hierarchy, and so $I\Delta_0$ is in some sense a reasonable theory to reason about such sets. Numerous papers concerning how much number theory and combinatorics can be done in $I\Delta_0$ have been published and the interested reader should consult Hájek and Pudlák [7] or Krajíček [9] both as introductions to this area and for references into the literature.

Since Buss [1] presented a theory S^1_2 for polynomial time, many bounded arithmetic theories have been proposed to model reasoning about a variety of complexity classes. In particular, Clote and Takeuti [3] present theories for a variety of complexity classes within polynomial time. One such theory is TLS. Clote and Takeuti show that the essentially sharply bounded predicates of TLS are precisely LOGSPACE. In a later paper [19], Takeuti shows that a subtheory of TLS is able to prove the consistency of Frege propositional proof systems. From the point of view of propositional complexity Frege systems are considered quite strong and at the time of this writing no nontrivial lower bounds on proof size for families of tautologies in these systems are known. Cook [2] a describes a potentially stronger proof system still, $L\text{-}Frege$, and shows the second-order theory of Zambella [21] for LOGSPACE can prove $L\text{-}Frege$'s consistency. It is quite likely that TLS can also prove $L\text{-}Frege$'s consistency.

The goal of this paper is to show that Nepomnjaščiĭ's Theorem has important implications for the provable consequences of $I\Delta_0$ and TLS. The results are presented using a conservative extension of $I\Delta_0$ known as S_1 and a variant on Clote and Takeuti's TLS which is in a language with multiplication and is axiomatized in a simpler fashion than their theory. The version of TLS used here contains Clote and Takeuti's, still has as its $\hat{\Delta}^b_1$-predicates LOGSPACE. Using Nepomnjaščiĭ's Theorem and Parikh's Theorem, it is shown that at least one of the theories S_1 and our TLS cannot prove that all Σ_1-sets are Diophantine (i.e., the Matiyasevich-Robinson-Davis-Putnam (MRDP) Theorem [11]). It was already known that $I\Delta_0{+}exp$, where exp is an axiom for exponentiation, proves the MRDP Theorem [6]. Being careful with how one defines a universal predicate for $\hat{\Sigma}^{\mathsf{b}}_{i,k}$-formulas, our paper also shows using Nepomnjaščiĭ's Theorem that TLS cannot prove $\hat{\Sigma}^{\mathsf{b}}_{1,1} = \hat{\Pi}^{\mathsf{b}}_{1,1}$. This is fairly close to saying (but not quite) that TLS cannot prove NLIN =co-NLIN. Using the techniques of Pollett and Pruim [16], it is possible that the latter result could be obtained with the techniques of this paper but the expense would be to make TLS a more awkward looking theory. The arguments presented for the results above can be generalized to where simply defined functions of quasi-linear growth are added to both TLS and S_1.

As a final point before proceeding to the outline of the paper, it should be noted that because of Parikh's Theorem, what the MRDP theorem is for a bounded arithemtic theory depends on the fastest growth rate functions in the underlying language. For instance, for $I\Delta_0$ to be able to prove MRDP, it suffices for it to show that linear sized bounded quantifiers can be eliminated in a Diophantine way. In the *TLS* case, since there is a function of growth rate $2^{|x||y|}$ in the language, one needs to be able to eliminate polynomial sized bounded quantifiers in a Diophantine way. Thus, the recent work in Pollett [15], which is in a language with 2^x is incomparable with the results of this paper.

This paper is organized as follows: The next section contains the notations and main definitions used in this paper. This is followed by a section showing that the $\hat{\Delta}^b_1$-predicates of *TLS* are in fact LOGSPACE. The first two results listed in the abstract are then presented.

2 Preliminaries

The language L_1 contains the non-logical symbols: 0, S, $+$, $\cdot$, $\leq$, $\dot{-}$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $\mathrm{PAD}(x, y)$, and $\mathrm{MSP}(x, i)$. The symbols 0, $S(x) = x + 1$, $+$, $\cdot$, and $\leq$ have the usual meaning. The intended meaning of $x \dot{-} y$ is x minus y if this is greater than zero and zero otherwise, $\lfloor \frac{1}{2}x \rfloor$ is x divided by 2 rounded down, and $|x|$ is $\lceil \log_2(x + 1) \rceil$, that is, the length of x in binary notation. $\mathrm{PAD}(x, y)$ is intended to mean $x \cdot 2^{|y|}$ and will be useful in defining a pairing function as an L_1-term. Finally, $\mathrm{MSP}(x, i)$ stands for 'most significant part' and is intended to mean $\lfloor x/2^i \rfloor$. The language L_2 is $L_1 \cup \{\#\}$. $x\#y$ reads 'x smash y' and is intended to mean $2^{|x||y|}$. The notation 1 is used for $S(0)$, 2 for $S(S(0))$, etc. A quantifier of the form $(\forall x \leq t)$ or $(\exists x \leq t)$ where t is a term not containing x is called a *bounded quantifier.* A formula is *bounded* or Δ_0 if all its quantifiers are. A quantifier of the form $(\forall x \leq |t|)$ or of the form $(\exists x \leq |t|)$ is called *sharply bounded* and a formula is *sharply bounded* if all its quantifiers are. Given a language L, the hierarchy of formulas $E_{i,L}$ and $U_{i,L}$ are defined as follows: $E_{1,L}$ are those formulas of the form $(\exists x \leq t)\phi$ and $U_{1,L}$ are those formulas of the form $(\forall x \leq t)\phi$ where ϕ is an open formula. $E_{i,L}$ are those formulas of the form $(\exists x \leq t)\phi$ where $\phi \in U_{i-1,L}$-formula. $U_{i,L}$ are those formulas of the form $(\forall x \leq t)\phi$ where $\phi \in E_{i-1,L}$. The notations E_i and U_i are used when L is understood, and $E_{i,k}$ and $U_{i,k}$ are used for E_{i,L_k} and U_{i,L_k}. The class of quantifier-free formulas is denoted by *open* (or $open_k$ to emphasize the language is L_k). For $i > 0$, a $\hat{\Sigma}^{\mathsf{b}}_i$-formula (resp. $\hat{\Pi}^{\mathsf{b}}_i$-formula) is defined to be a E_{i+1}-formula (resp. U_{i+1}-formula) whose innermost quantifier is sharply bounded. To emphasize the language is L_k we write $\hat{\Sigma}^{\mathsf{b}}_{i,k}$ and $\hat{\Pi}^{\mathsf{b}}_{i,k}$. The classes Σ^{b}_i and Π^{b}_i are the closures of $\hat{\Sigma}^{\mathsf{b}}_i$ and $\hat{\Pi}^{\mathsf{b}}_i$ under subformulas, $\wedge$, $\vee$, and sharply bounded quantifications. Kent and Hodgson [8] (see also Pollett [17]) have shown the sets defined by $\hat{\Sigma}^{\mathsf{b}}_{i,2}$-(resp. $\hat{\Pi}^{\mathsf{b}}_{i,2}$-)formulas are precisely the Σ^p_i-(resp. Π^p_i-)predicates. Thus, the $\hat{\Sigma}^{\mathsf{b}}_{1,2}$-formulas correspond to the NP-predicates.

The theory $BASIC_k$ is axiomatized by all substitution instances of a finite set of quantifier free axioms for the non-logical symbols of L_k, $k = 1, 2$. These are listed in Buss [1] except for the axioms for MSP and $\dot{-}$ which are listed in Takeuti [18], and those for PAD are listed in Clote and Takeuti [3].

For this paper, it is useful to be able to have a pairing function, as well as to have functions that can project blocks of bits from a number so that a limited amount of sequence coding can be done. These can be defined using L_1-terms as follows: For projection of bits, define the functions $2^{|y|} := \mathrm{PAD}(1, y)$, $2^{\min(|y|,x)} := \mathrm{MSP}(2^{|y|}, |y| \dot{-} x)$, $\mathrm{LSP}(x, i) := x \dot{-} \mathrm{MSP}(x, i) \cdot 2^{\min(|x|,i)}$, $\hat{\beta}_{|t|}(x, w) := \mathrm{MSP}(\mathrm{LSP}(w, (Sx)|t|), x|t|)$, and $\mathrm{BIT}(i, x) := \hat{\beta}_1(i, x)$. Here $\hat{\beta}$ is supposed to project the xth block of $|t|$ bits from w and BIT is supposed to return the ith bit of x. Given these functions to define pairing operations, let $\max(x, y) := (1 \dot{-} ((x + 1) \dot{-} y)))y + (1 \dot{-} (y \dot{-} x))x$ and set $B = 2^{|\max(x,y)|+1}$. Thus, B will be longer than either x or y. Define an ordered pair as $\langle x, y\rangle := (2^{|\max(x,y)|} + y) \cdot B + (2^{|\max(x,y)|} + x)$. To project out the coordinates from such an ordered pair, use $(w)_1 := \hat{\beta}_{\lfloor \frac{1}{2}|w|\rfloor \dot{-} 1}(0, \hat{\beta}_{\lfloor \frac{1}{2}|w|\rfloor}(0, w))$ and $(w)_2 := \hat{\beta}_{\lfloor \frac{1}{2}|w|\rfloor \dot{-} 1}(0, \hat{\beta}_{\lfloor \frac{1}{2}|w|\rfloor}(1, w))$ which return the left and right coordinates of the pair w. To check if w is a pair the formula $ispair(w) :=$

$$Bit(w, \lfloor \frac{1}{2}|w|\rfloor \dot{-} 1) = 1 \wedge 2 \cdot |\max((w)_1, (w)_2)| + 2 = |w|$$

is used. The usual properties of this formula as well as the terms listed above are provable in the theories we will consider in this paper [17].

The theories in this paper will all be formulated in the sequent calculus system LKB of Buss [1].

Definition 1 *A Ψ-$L^m IND$ inference is an inference:*

$$\frac{A(b), \Gamma \rightarrow A(Sb), \Delta}{A(0), \Gamma \rightarrow A(|t(x)|_m), \Delta}$$

where b is an eigenvariable and must not appear in the lower sequent, $t \in L_2$, $|x|_0 = x$, and $|x|_{m+1} = ||x|_m|$.

The notations IND, $LIND$, $LLIND$ will be used instead of $L^0 IND$, $L^1 IND$, and $L^2 IND$.

Definition 2 ($i \geq 0$) *The theories T^i_k and S^i_k are $BASIC_k$+$\hat{\Sigma}^{\mathsf{b}}_{i,k}$-IND and $BASIC_k$+$\hat{\Sigma}^{\mathsf{b}}_{i,k}$-LIND, respectively.*

We define $S_k := \cup_i S^i_k$.

That S^i_k and T^i_k can be equivalently defined using $\hat{\Sigma}^{\mathsf{b}}_{i,k}$ induction schemas rather than $\Sigma^{\mathsf{b}}_{i,k}$ schemas was shown in Pollett [17]. From Buss [1] is it known that

$$S^i_k \subseteq T^i_k \subseteq S^{i+1}_k.$$

The theory $I\Delta_0$ is defined using the language $0,S,+,\cdot;\leq$. It consists of axioms for these symbols together with Δ_0-*IND*. The symbols in L_1 are all definable in $I\Delta_0$, and it is known that S_1 is a conservative extension of $I\Delta_0$. For more details on this relationship and this theory, the reader is advised to consult Krajíček [9].

The last definitions needed to present *TLS* are now given.

Definition 3 *Given a term t in one of the languages of this paper we define a monotonic term t^* as follows: If t is constant or a variable, then $t = t^*$. If t is $f(s)$, where f is a unary function symbol, then t^* is $f(s^*)$. If t is $s_1 \circ s_2$ for $\circ$ a binary operation other than $\dot{-}$ or MSP, then t^* is $s_1^* \circ s_2^*$. Lastly, if t is $s_1 \dot{-} s_2$ or $\mathrm{MSP}(s_1, s_2)$, then t^* is s_1^*.*

It is easily proved in $BASIC$ + *open-LIND* that t^* is monotonic, and $t \leq t^*$.

In the next definition, $\exists!$ is used to abbreviate two sequents expressing uniqueness and existence.

Definition 4 *The Ψ-WSN (weak successive nomination rule) is the following rule:*

$$\frac{b \leq |k(j,\vec{a})| \rightarrow \exists! x \leq |k| A(j,\vec{a},b,x)}{\rightarrow \exists w \leq \mathrm{bd}(|k|,t) \forall j < |t| A(j,\vec{a},\hat{\beta}_{|k^*|}(j,w),\hat{\beta}_{|k^*|}(Sj,w))}$$

where $A \in \Psi$ and $\mathrm{bd}(a,b) := 2(2a\#2b)$.

The last rule needed to define *TLS* is:

Definition 5 *Ψ-REPL (quantifier replacement) is the following rule:*

$$(\forall x \leq |s|)(\exists y \leq t(x,a))A(x,y,a) \Leftrightarrow$$
$$(\exists w \leq \mathrm{bd}(t^*(|s|,a),s))(\forall x \leq |s|)A(x,\dot{\beta}_{|t^*(|s|,a)|,t}(x,w))$$

where $A \in \Psi$ and $\dot{\beta}_{t,s}(x,w) := \min(\hat{\beta}_t(x,w),s)$. Here $\min(x,y) := x + y \dot{-} \max(x,y)$.

Definition 6 *TLS is $BASIC_2$+$open_2$-LIND+$\hat{\Sigma}^{\mathsf{b}}_{1,2}$-WSN+$\hat{\Sigma}^{\mathsf{b}}_{1,2}$-REPL.*

3 Bootstrapping

TLS is axiomatized in a different fashion than the version presented in Clote and Takeuti [3]. The theory here actually has a slightly stronger axiomatization. Nevertheless, in this section it is argued that its $\hat{\Delta}^b_1$-predicates are still LOGSPACE.

Recall A is said to be $\hat{\Delta}^b_i$ *in a theory* T if $T \vdash A^\Sigma \equiv A \equiv A^\Pi$ where A^Σ is $\hat{\Sigma}^{\mathsf{b}}_i$ and A^Π is $\hat{\Pi}^{\mathsf{b}}_i$. Δ^b_i is defined analogously, but using Σ^{b}_i and Π^{b}_i. Recall also that f is Ψ*-defined in* T if there is a Ψ-formula A such that $\mathbb{N} \models A(x, f(x))$ and $T \vdash \forall x \exists! y A(x, y)$. Because *TLS* proves quantifier replacement for $\hat{\Sigma}^{\mathsf{b}}_1$-formulas, the notions of Σ^{b}_1-definability and $\hat{\Sigma}^{\mathsf{b}}_1$-definability coincide; similarly, the notions $\hat{\Delta}^b_1$ and Δ^b_1-coincide.

Johannsen and Pollett [10] give two theories for the TC^0-predicates (predicates computable by constant depth threshold circuits), C^0_2 and Δ^b_1-CR. The former theory is of interest in the present discussion. It was axiomatized as $BASIC$+$open_2$-$LIND$ and Σ^{b}_0-$REPL$ and so is contained in *TLS*. This is because it is easy to show Σ^{b}_0-$REPL$ and in fact even Σ^{b}_1-$REPL$ using $\hat{\Sigma}^{\mathsf{b}}_1$-$REPL$. Given that Σ^{b}_0-$REPL$ implies Σ^{b}_1-$REPL$ by the same method as was used in Buss [1] to show Π^{b}_i-$REPL$ implies $\Sigma^{\mathsf{b}}_{i+1}$-$REPL$, the only difference between C^0_2 and *TLS* is that the latter theory has $\hat{\Sigma}^{\mathsf{b}}_{1,2}$-$WSN$. In what follows, a function is said to be in TC^0 or in LOGSPACE, if its graph is in the given class and if the number of bits in its output is polynomial in the number of input bits.

Theorem 1 *(1) The* $\hat{\Sigma}^{\mathsf{b}}_1$*-definable functions of TLS are exactly* LOGSPACE*. (2) The* $\hat{\Delta}^b_1$*-predicates of TLS are exactly the* LOGSPACE *predicates.*

Proof. From Clote and Takeuti [3], the functions in LOGSPACE can be viewed as the closure TC^0 under B_2RN. Here a function f is defined by B_2RN from the functions g, h_0, h_1, and k if $f(0, \vec{x}) = g(\vec{x})$, $f(2n, \vec{x}) = h_0(n, \vec{x}, f(n, \vec{x}))$, $f(2n + 1, \vec{x}) = h_1(n, \vec{x}, f(n, \vec{x}))$ and, in addition, it is required that $f(n, \vec{x}) < |k(n, \vec{x})|$. Given that *TLS* contains C^0_2 and Johannsen and Pollett [10] show the Σ^{b}_1-definable functions of C^0_2 are precisely TC^0, it follows *TLS* can Σ^{b}_1-define TC^0. Using $\hat{\Sigma}^{\mathsf{b}}_1$-$REPL$, it can thus $\hat{\Sigma}^{\mathsf{b}}_1$-define these functions. Then by using $\hat{\Sigma}^{\mathsf{b}}_1$-$WSN$ and essentially same argument as used in Theorem 5.1 and 6.3 by Clote and Takeuti [3] for their version of *TLS*, one can show *TLS* can proves it $\hat{\Sigma}^{\mathsf{b}}_1$-definable functions closed under B_2RN. For the other direction, one needs to carry out a Buss-style witnessing argument, to show that only the LOGSPACE functions are $\hat{\Sigma}^{\mathsf{b}}_1$-definable by *TLS*. This argument is essentially the same as the witnessing argument of Johannsen and Pollett [10] to show C_2^0 can only $\hat{\Sigma}^{\mathsf{b}}_1$-define TC^0 functions. The only additional case is to handle $\hat{\Sigma}^{\mathsf{b}}_1$-*WSN*. The witness function in this case is constructed using B_2RN in a similar fashion to Theorem 5.2 of Clote and Takeuti. The reader interested in more of the gory details can consult the technical report Pollett [14]. Given that the $\hat{\Sigma}^{\mathsf{b}}_1$-definable functions of *TLS* are those functions in $\mathsf{LOGSPACE}$, the fact that the $\hat{\Delta}^b_1$-predicates of *TLS* are exactly $\mathsf{LOGSPACE}$, follows from the usual correspondence between 0-1 valued $\hat{\Sigma}^{\mathsf{b}}_1$-definable functions and the $\hat{\Delta}^b_1$-predicates of a theory. This argument can be found in Buss [1]. □

4 Independence results

To begin some well known results are recalled:

Theorem 2 *(1) The predicates in* $\cup_i\hat{\Sigma}^{\mathsf{b}}_{i,1}$ *are precisely* LinH. *(Wrathall [20])*
(2) For $i > 0$, $\hat{\Sigma}^{\mathsf{b}}_{i,2} = \Sigma^{\mathsf{p}}_i$. *(Kent-Hodgson [8])*

Theorem 3 *(Nepomnjaščiĭ [12])* LinH *contains* $\mathsf{TISP}(n^k, n^{1-\epsilon})$. *So* LinH *contains* $\mathsf{LOGSPACE}$.

The next lemma provides a universal predicate for $\hat{\Sigma}^{\mathsf{b}}_i$-formulas which will be convenient to work with in the sequel.

Lemma 1 *There is a* $\hat{\Sigma}^{\mathsf{b}}_{i,1}$*-formula (note the* 1*),* $U_i(e, x, z)$*, such that for any* $\hat{\Sigma}^{\mathsf{b}}_{i,2}$*-formula* $A(x)$ *there is a number* e_A *and* L_2*-term* t_A *for which*

$$TLS \vdash U_i(e_A, x, t_A(x)) \equiv A(x).$$

If A *is in* $\hat{\Sigma}^{\mathsf{b}}_{i,1}$ *then* t_A *can be chosen to be an* L_1*-term in* x *or we can choose a single* L_2*-term* $t(e_A, x)$ *which works for all* A.

Proof. Using $\mathrm{K}_{\neg}(x) := 1 \dot{-} x$, $\mathrm{K}_{\vee}(x, y) := x + y$, and $\mathrm{K}_{\leq}(x, y) := \mathrm{K}_{\neg}(y \dot{-} x)$, one can write any open formula $A(x, \vec{y})$ as an equation $f(x, \vec{y}) = 0$ where $f \in L_k$. By induction, on the complexity of A this is provable in *TLS*. So any $\hat{\Sigma}^{\mathsf{b}}_i$-formula $\phi(x)$ is provably equivalent in *TLS* to one of the form

$$(\exists y_1 \leq t_1) \cdots (Qy_i \leq t_i)(Q' y_{i+1} \leq |t_{i+1}|)(t_{i+2}(x, \vec{y}) = 0)$$

where the quantifiers Q and Q' will depend on whether i is even or odd. We fix some coding scheme for the 12 symbols of L_2 as well as for the $i+2$ variables $x, y_1, \ldots, y_{i+1}$. We use $\ulcorner\urcorner$ to denote the code for some symbol. i.e.,

$\ulcorner = \urcorner$ is the code for $=$. We choose our coding so that all codes require less than $|i+14|$ bits and 0 is used as $\ulcorner NOP \urcorner$ meaning no operation. Thus, if one tries to project out operations beyond the end of the code of the term one naturally just projects out $\ulcorner NOP \urcorner$'s. The code for a term t is a sequence of blocks of length $|i+14|$ that write out t in postfix order. So $x + y_1$ would be coded as the three blocks $\ulcorner x \urcorner \ulcorner y_1 \urcorner \ulcorner + \urcorner$. The code for a $\hat{\Sigma}^{\mathsf{b}}_i$-formula will be $\langle\langle \ulcorner t_1 \urcorner, \ldots, \ulcorner t_{i+3} \urcorner \rangle\rangle$. We now describe $U_i(e, x, z)$. It will be obtained from the formula

$$(\exists w \leq z)(\exists y_1 \leq z)(\forall j \leq |e|)(\forall y_2 \leq z) \cdots \\ \cdots (Qy_i \leq z)(Q' y_{i+1} \leq |z|)\phi_i(e, j, x, \vec{y})$$

after pairing is applied. Here ϕ_i consists of a statement saying w is a tuple of the form $\langle\langle w_1, \ldots, w_{i+2} \rangle\rangle$ together with statements saying each w_i codes a postfix computation of t_i in $e = \langle\langle \ulcorner t_1 \urcorner, \ldots, \ulcorner t_{i+3} \urcorner \rangle\rangle$. If $z' := MSP(z, \lfloor \frac{1}{2}|z| \rfloor)$ (roughly, the square root of z) is used as the block size, this amounts to checking conditions for each m

$$[\hat{\beta}_{|i+14|}(j, \ulcorner t_m \urcorner) = \ulcorner x \urcorner \supset \hat{\beta}_{|z'|}(j, w_m) = x] \wedge$$

$$[\hat{\beta}_{|i+14|}(j, \ulcorner t_m \urcorner) = \ulcorner + \urcorner \supset \\ \hat{\beta}_{|z'|}(j, w_m) = \hat{\beta}_{|z'|}(j \dot{-} 2, w_m) + \hat{\beta}_{|z'|}(j \dot{-} 1, w_m)] \wedge \cdots$$

$$[\hat{\beta}_{|i+14|}(j, \ulcorner t_m \urcorner) = \ulcorner \# \urcorner \supset \\ |\hat{\beta}_{|z'|}(j, w_m)| = S(|\hat{\beta}_{|z'|}(j \dot{-} 2, w_m)||\hat{\beta}_{|z'|}(j \dot{-} 1, w_m)|) \\ \wedge LSP(\hat{\beta}_{|z'|}(j, w_m), |\hat{\beta}_{|z'|}(j, w_m)| \dot{-} 1) = 0] \wedge \cdots$$

$$\cdots$$

$$[\hat{\beta}_{|i+14|}(j, \ulcorner t_m \urcorner) = \ulcorner NOP \urcorner \supset \hat{\beta}_{|z'|}(j, w_m) = \hat{\beta}_{|z'|}(j \dot{-} 1, w_m)].$$

ϕ_i also has conditions $y_m \leq \hat{\beta}_{|z'|}(|e|, w_m) \wedge$ if y_m was existentially quantified and conditions $y_m \leq \hat{\beta}_{|z'|}(|e|, w_m) \supset$ if y_m was universally quantified. Notice none of the conditions above make use of the $\#$ function. Finally, ϕ_i has a condition saying $\hat{\beta}_{|z'|}(|e|, w_{i+2}) = 0$. Since TLS can prove simple facts about projections from pairs, it can prove by induction on the complexity of the terms in any $\hat{\Sigma}^{\mathsf{b}}_i$-formula $\phi(x)$ that $U_i(e_\phi, x, t(e_\phi, x)) \equiv \phi(x)$ provided $t(e_\phi, x)$ is large enough.

To estimate the size of t_A, an upper bound on w_m is calculated. First, all real formulas A have their terms represented as trees, so we can assume e_A codes terms which are trees. By induction over the subtrees of a given term

t_m, one can show an upper bound on the block size needed to store a step of w_m of the form $|e_m|(|x|+|e_A|)$. So the length of any w_m can be bounded by $\ell = |e_A||e_A|(|x|+|e_A|) > |e_m||e_m|(|x|+|e_A|)$. So choosing an L_1-term larger than $2^{(i+2)\ell}$ suffices. This is possible since e_A is a fixed number. Notice if both e_A and x are viewed as parameters, this is in fact boundable by an L_2-term t. If A does involve # than a similar estimate can be done to show that an L_2-term for t_A suffices. □

Lemma 2 *For $i \geq 1$, $\hat{\Sigma}^{\mathsf{b}}_{i,1} \neq \hat{\Pi}^{\mathsf{b}}_{i,2}$.*

Proof. If A is in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$ then the last argument of U_i from Lemma 1 is an L_2-term. So there is a $\hat{\Sigma}^{\mathsf{b}}_{i,2}$-formula $U(x, e_A) \equiv A$ for all A in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$. Consider $\neg U(x, x)$ this formula is equivalent to a $\hat{\Pi}^{\mathsf{b}}_{i,2}$-formula. Also, it is easy to see it is not in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$. □

The independence results in this section are all a consequence of the following lemma:

Lemma 3 *If $\hat{\Sigma}^{\mathsf{b}}_{i,1} = \hat{\Pi}^{\mathsf{b}}_{i,1}$ then* $\mathsf{LOGSPACE} \neq \mathsf{NP}$.

Proof. Suppose $\hat{\Sigma}^{\mathsf{b}}_{i,1} = \hat{\Pi}^{\mathsf{b}}_{i,1}$ and $\mathsf{LOGSPACE} = \mathsf{NP}$. As $\mathsf{LOGSPACE}$ is closed under complement $\mathsf{LOGSPACE} = \mathsf{PH}$. By Theorem 3 and Theorem 2 $\mathsf{LOGSPACE}$ is contained in LinH, and we have that $\hat{\Sigma}^{\mathsf{b}}_{i,1} = \mathsf{LinH} = \mathsf{PH}$. But by Lemma 2, there are languages in $\hat{\Pi}^{\mathsf{b}}_{i,2}$ that are not in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$. □

Lemma 3 is similar to a result of Ferreira [4] where it is shown that $\mathsf{LOGSPACE} = \Delta_0$ implies $\Delta_0 \not\subseteq \Sigma^l_s$. Here Σ^l_s is a secord-order class of formulas defining sets similar to $\Sigma^{\mathsf{b}}_{s,1}$. Ferreira's argument was model theoretic. One consequence of Lemma 3 concerns the provability of the Matiyasevich-Robinson-Davis-Putnam (MRDP) Theorem [11] in bounded arithmetic. Recall the MRDP Theorem says that the Σ_1-sets are equivalent to the sets that can be defined by formulas of the form:

$$A = \{x | (\exists \vec{y}) P(x, \vec{y}) = Q(x, \vec{y})\},$$

where P, Q are polynomials with coefficients in $\mathbb{N}$. It is known that $I\Delta_0 + exp$, where exp is an axiom for exponentiation, proves the MRDP Theorem [6]. To prove our result, we first have need of a well-known lemma whose proof we include for completeness.

Lemma 4 *Let T be one of S^i_k, S_k or TLS. If T proves the MRDP theorem then T proves $E_{1,k} = U_{1,k}$.*

Proof. To see this, suppose T proves the MRDP theorem. Then for every $U_{1,k}$-formula $A(\vec{x})$ there is a formula $F(\vec{x}) := (\exists\vec{y})P(\vec{x},\vec{y}) = Q(\vec{x},\vec{y})$ where P, Q are polynomials such that $T \vdash A \equiv F$. In particular, T proves $A \rightarrow (\exists\vec{y})P(\vec{x},\vec{y}) = Q(\vec{x},\vec{y})$. By Parikh's theorem (see Hájek and Pudlák [7] for a proof), since T is a bounded theory one can bound the $\vec{y}$'s by an L_k-term t giving an $E_{1,k}$-formula F_2. Note $F_2 \supset F \supset A$ so $A \equiv F_2$ completing the proof. □

Theorem 4 *At least one of* S_1 *and* TLS *does not prove MRDP.*

Proof. By the previous lemma, if S_1 proves the MRDP Theorem then $\mathsf{LinH} = \hat{\Sigma}^{\mathsf{b}}_{1,1}$. By a similar, argument if TLS proves MRDP Theorem then $\mathsf{LOGSPACE} = \hat{\Pi}^{\mathsf{b}}_{1,2} = \hat{\Sigma}^{\mathsf{b}}_{1,2} = \mathsf{PH}$. Thus, we contradict Lemma 3. □

The next theorem gives another application of Lemma 3.

Theorem 5 TLS *cannot prove* $\hat{\Sigma}^{\mathsf{b}}_{1,1} = \hat{\Pi}^{\mathsf{b}}_{1,1}$.

Proof. Suppose TLS proves $\hat{\Sigma}^{\mathsf{b}}_{1,1} = \hat{\Pi}^{\mathsf{b}}_{1,1}$. This means that for each $\hat{\Sigma}^{\mathsf{b}}_{1,1}$-formula A we can find some $\hat{\Pi}^{\mathsf{b}}_{1,1}$-formula B such that $TLS \vdash A \equiv B$. Let $A(x) := \exists y \leq t(x) D(x,y)$ be an arbitrary $\hat{\Sigma}^{\mathsf{b}}_{1,2}$-formula in one variable. Let $C(x,z) := U_1(e_A, x, z)$ where U_1 is from Lemma 1. So C is a $\hat{\Sigma}^{\mathsf{b}}_{1,1}$-formula, and, thus, by assumption, provably equivalent to some $\hat{\Pi}^{\mathsf{b}}_{1,1}$-formula $C'(x,z)$ in TLS. So TLS proves

$$A \equiv C(x, t_A(x)) \equiv C'(x, t_A(x))$$

where t_A is the bounding term on U_1 in Lemma 1. The last formula is a $\hat{\Pi}^{\mathsf{b}}_{1,2}$-formula. Hence, it follows that TLS proves $\hat{\Sigma}^{\mathsf{b}}_{1,2} = \hat{\Pi}^{\mathsf{b}}_{1,2}$. i.e., $\mathsf{NP} = \mathsf{co}$-$\mathsf{NP}$. As the $\hat{\Delta}^{b}_{1}$-formulas of TLS are $\mathsf{LOGSPACE}$, one also gets that $\hat{\Sigma}^{\mathsf{b}}_{1,2} = \mathsf{LOGSPACE}$. But this contradicts Lemma 3. □

Remark 1 *The results presented above are reasonably insensitive to the underlying language as long as the functions symbols added are* $\mathsf{LOGSPACE}$ *computable and have* $O(n^{1+o(1)})$ *growth rate. For instance, one could add to* L_1 *and* L_2 *a symbol for* $x\#|y|$ *and add to* S_1 *and* TLS *defining axioms for this symbol. The resulting* TLS *would be conservative over the* TLS *used above. On the hand, the* Δ_0*-sets in the resulting* L_1 *would now define the quasi-linear time hierarchy and the resulting* S_1 *would be able to reason about such sets. Nevertheless, the part of Lemma 1 concerning a single* L_2*-term able to* *work for all A still holds. Now, though, a bound on the length of the code for computation of* e_A *will be* $2^{(i+2)\ell}$ *where* ℓ *is* $O((|x|+|e_A|)(||x|+|e_A||)^{|e_A|})$. *If one requires that* $e_A \leq ||x||$ *then strings of this length can be bounded by an* L_1*-term. So in Lemma 2, one now considers a* $\hat{\Pi}^{\mathsf{b}}_{1,2}$ *predicate* $\neg U(x, ||x||)$ *to diagonalize out of* $\hat{\Sigma}^{\mathsf{b}}_{1,1}$. *All the other results of this section also hold. Hence, it still holds that at least one of* S_1 *or TLS in the new languages does not prove MRDP and also that TLS does not prove* $\hat{\Sigma}^{\mathsf{b}}_{1,1} = \hat{\Pi}^{\mathsf{b}}_{1,1}$.

5 Conclusion

Hájek and Pudlák [7] develop definitions for context free grammars in the theory $I\Delta_0$. Thus, it is quite likely that the results of this paper could be extended to a theory whose Δ^b_1-predicates were LOGCFL. Here LOGCFL is the class of languages logspace reducible to context free languages. It is known that LOGCFL contains NLOGSPACE. So such a result seems like the next logical step in pushing the techniques of this paper.

References

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

[2] S. Cook. A Survey of Complexity Classes and their Associated Propositional Proof Systems and Theories, and a Proof System for Log Space Slides for Edinburgh talk, presented at the ICMS Workshop "Circuit and Proof Complexity", Edinburgh, October, 2001.

[3] P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 154–218. Birkhäuser, Boston, 1995.

[4] F. Ferreira. A proof that LOGSPACE $\neq$ NLIN. Unpublished notes. 1998.

[5] L. Fortnow Time-space tradeoffs for satisfiability. Journal of Computer and System Sciences, 60(2):337-353, April 2000.

[6] H. Gaifman and C. Dimitracopoulos. Fragments of Peano's arithmetic and the MRDP theorem. Monographie 30 de L'Enseignement Mathématique, pages 187–206, 1982.

[7] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetics.* Springer-Verlag, 1993.

[8] C. F. Kent and B.R. Hodgson. An arithmetical characterization of NP. *Theoretical Computer Science*, 21:255–267, 1982.

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

[10] J. Johannsen and C. Pollett. On the Δ_1^b-Comprehension Rule. In S. Buss, P. Hájek and P. Pudlák *Lecture Notes in Logic 13 – Logic Colloquium 1998*, pages 269–286, A.K. Peters, 2000.

[11] Y. Matiyasevich. Enumerable sets are Diophantine. *Dokl. Acad. Nauk*, 191:279–282, 1970.

[12] V.A. Nepomnjaščiĭ. Rudimentary predicates and Turing computations. *Dokl. Acad. Nauk*, Vol. 195, pages 282–284, 1970, transl. Vol. 11 1462–1465, 1970.

[13] C. Pollett. Multifunction algebras and the provability of $\mathsf{PH}\downarrow$. Annals of Pure and Applied Logic. Vol. 104 July 2000. pp. 279–303.

[14] C. Pollett. Nepomnjaščiĭ's Theorem and Independence Proofs in Bounded Arithmetic. Electronic Colloquium on Computational Complexity. TR02-051.

[15] C. Pollett. $S_{k,exp}$ does not prove $NP = co\text{-}NP$ uniformly. Submitted.

[16] C. Pollett and R. Pruim. Strengths and Weaknesses of LH Arithmetic. *Mathematical Logic Quarterly.* 48:221–243(No.2) Feb. 2002.

[17] C. Pollett. Structure and definability in general bounded arithmetic theories. *Annals of Pure and Applied Logic.* Vol. 100. pages 189–245, October 1999.

[18] G. Takeuti. $RSUV$ isomorphisms. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, volume 23 of *Oxford Logic Guides*, pages 364–386. Clarendon Press, Oxford, 1993.

[19] G. Takeuti. Frege Proof Systems and TNC^0. *The Journal of Symbolic Logic*, 63(2):709-738, June 1998.

[20] C. Wrathall. Rudimentary Predicates and Relative Computation. *SIAM Journal of Computing*, Vol. 7 pp. 194–209, 1978.

[21] D. Zambella. End Extensions of Models of Linearly Bounded Arithmetic. Annals of Pure and Applied Logic 88(2-3):263–277, 1997.