Nepomnjaščiĭ's Theorem and Independence Proofs in Bounded Arithmetic

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

February 20, 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, the following statements are proven: (1) At least one of S_1 or *TLS* does not prove the Matiyasevich-Davis-Robinson-Putnam Theorem and (2) *TLS* does not prove $\hat{\Sigma}^{\mathsf{b}}_{1,1} = \hat{\Pi}^{\mathsf{b}}_{1,1}$. Here S_1 is a conservative extension of the well-studied theory $I\Delta_0$ and *TLS* is a theory whose $\hat{\Delta}^b_{1,2}$-predicates are precisely $\mathsf{LOGSPACE}$. The relation of *TLS* from this paper to previously studied theories is also developed and generalizations of the previous two results to quasi-linear settings are discussed as well.

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

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 [19] 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 Pudák [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 [18], 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-$Frege$, and shows the second-order theory of Zambella [20] for LOGSPACE can prove L-$Frege$'s consistency. It is quite likely that TLS can also prove L-$Frege$'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, and is Σ^{b}_1-conservative over their theory. 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 [15], 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.

A lesser goal of this paper is to clarify the relationship between the *TLS* of this paper and Clote and Takeuti's theory. Simplifications to Clote and Takeuti's theories for $AC^0(2)$, $AC^0(6)$, and NC^1 are also briefly discussed. The Σ^{b}_1-conservativety result between the theory of this paper and Clote and Takeuti's is very much like a first-order version of Zambella's result [20] concerning the Σ^{p}_1-conservativety of Σ^{p}_1-(rec+choice) over Σ^{p}_0-*rec*. In fact, Zambella's paper mentions that he is is unaware of any first-order theory corresponding to his second-order theories for LOGSPACE. Given the results of this paper it seems quite likely that Clote and Takeuti's and our theories play this role.

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 $\hat{\Delta}^b_{1,2}$-predicates of *TLS* are in fact LOGSPACE and that the *TLS* of this paper is Σ^{b}_1-conservative over the *TLS* of Clote and Takeuti [3]. The first two results listed in the abstract are then presented.

2 Preliminaries

The language L_0 contains the non-logical symbols: 0, S, $+$, $=$, $\leq$, $\dot{-}$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $\mathrm{PAD}(x,y)$, and $\mathrm{MSP}(x,i)$; the language L_1 is $L_0 \cup \{\cdot\}$. 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 functions and projections using just L_0-terms. 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 $L_1 \cup \{\#\}$ and $L_0^{\#}$ is $L_0 \cup \{\#\}$. $x\#y$ 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 [13]) 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 subtitution instances of a finite set of quantifier free axioms for the non-logical symbols of L_k, $k = 0, 1, 2$. These are listed in Buss [1] except for the axioms for MSP and $\dot{-}$ which are listed in Takeuti [17], and those for PAD are listed in Clote and Takeuti [3]. Some useful L_0-terms are listed below:

$$
\begin{array}{rcl}
2^{|y|} & := & \mathrm{PAD}(1, y) \\
2^{\min(|y|,x)} & := & \mathrm{MSP}(y, |y| \dot{-} x) \\
\mathrm{K}_\neg(x) & := & 1 \dot{-} x \\
\mathrm{K}_\vee(x, y) & := & x + y \\
\mathrm{K}_\leq(x, y) & := & \mathrm{K}_\neg(y \dot{-} x) \\
\mathrm{LSP}(x, i) & := & x \dot{-} \mathrm{PAD}(\mathrm{MSP}(x, i), i) \\
\mathrm{DMSB}(x) & := & \mathrm{LSP}(x, |x| \dot{-} 1) \\
\mathrm{mod2}(x) & := & x \dot{-} \mathrm{PAD}(\lfloor \frac{1}{2} x \rfloor, 1) \\
\mathrm{BIT}(i, x) & := & \mathrm{mod2}(MSP(x, i)) \\
\mathrm{cond}(x, y, z) & := & \mathrm{MSP}(y, \mathrm{PAD}(\mathrm{K}_\neg(x), y)) + \mathrm{MSP}(z, \mathrm{PAD}(\mathrm{K}_\neg(\mathrm{K}_\neg(x)), z)) \\
\max(x, y) & := & \mathrm{cond}(\mathrm{K}_\leq(x, y), y, x) \\
\min(x, y) & := & \mathrm{cond}(\mathrm{K}_\leq(x, y), x, y)
\end{array}
$$

The following $L^{\#}_0$ terms will also sometimes be used:

$$
\begin{array}{rcl}
(b \cdot c)_{|a|} & := & \mathrm{cond}(\mathrm{K}_\vee(\mathrm{K}_\neg(b), \mathrm{K}_\neg(c)), 0, 2^{\min(|a|, b \dot{-} 1)} \# 2^{\min(|a|, c \dot{-} 1)}) \\
\hat{\beta}_t(x, w) & := & \mathrm{LSP}(\mathrm{MSP}(w, (x \cdot t)_{|w|}), t) \\
\dot{\beta}_{t,s}(x, w) & := & \min(\hat{\beta}_t(x, w), s)
\end{array}
$$

For brevity, this paper uses $2^{\ell(x)}$ for $2^{\min(|t(x)|, \ell(x))}$, if $\ell(x)$ is a term which is obviously less than some $|t(x)|$.

As for the intended meaning of some of the terms above, $\hat{\beta}_t(x, w)$ projects out the xth block (starting with a 0th block) of t bits from w. $\dot{\beta}_{t,s}(x, w)$

returns the minimum of $\hat{\beta}_t(x,w)$ and s. Note if the language were L_1, as is the case for Lemma 6 latter in this paper, one can use the usual '$\cdot$' rather than $(b\cdot c)_{|a|}$ to define $\hat{\beta}$ and $\dot{\beta}$. A term like $\hat{\beta}$ that projects blocks that are powers of 2 is also definable in L_0, but would make the correspondence between the theories of this paper and Clote and Takeuti's harder to establish.

The pairing operation which will be used is defined as follows. Let $B = 2^{|\max(x,y)|+1} = \mathrm{PAD}(2, \max(x,y))$. 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 $(w)_1 := \mathrm{DMSB}(\mathrm{LSP}(w, \lfloor\frac{1}{2}|w|\rfloor))$ and $(w)_2 := \mathrm{DMSB}(\mathrm{MSP}(w, \lfloor\frac{1}{2}|w|\rfloor))$ which return the right and left coordinates of the pair w. To check if w is a pair the function $\mathrm{ispair}(w) :=$

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

is used. Notice the above functions can all be expressed as L_0-terms and this last predicate can be expressed as an open L_0-formula.

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^0IND, L^1IND, and L^2IND.

Definition 2 *($i \geq 0$) The theories T^i_k and S^i_k axiomatized as $BASIC_k$+$\hat{\Sigma}^{\mathsf{b}}_{i,k}$-$IND$ 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 [13]. From Buss [1] is it known that

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

It should be noted that the correctness of the pairing function given above can be proven using the techniques of Pollett [13] in $BASIC_0$+$open_0$-$LIND$.

The theory $I\Delta_0$ is defined using the language $0,S,+,\cdot;\leq$. It consists of some base 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 the 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].

To introduce TLS, a function algebra characterization of Clote and Takeuti [3] for the functions in LOGSPACE needs to be discussed.

Definition 3 *The function f is defined by CRN from the functions g, h_0, and h_1 if*

$$\begin{aligned} f(0,\vec{x}) &= g(\vec{x}) \\ f(2n,\vec{x}) &= 2f(n,\vec{x}) + h_0(n,\vec{x}) \\ f(2n+1,\vec{x}) &= 2f(n,\vec{x}) + h_1(n,\vec{x}) \end{aligned}$$

Definition 4 *The function f is defined by B_2RN from the functions g, h_0, h_1, and k if*

$$\begin{aligned} f(0,\vec{x}) &= g(\vec{x}) \\ f(2n,\vec{x}) &= h_0(n,\vec{x},f(n,\vec{x})) \\ f(2n+1,\vec{x}) &= h_1(n,\vec{x},f(n,\vec{x})) \end{aligned}$$

provided that $f(n,\vec{x}) < |k(n,\vec{x})|$.

Theorem 1 *The functions in* LOGSPACE *are precisely those functions containing the $L_0^{\#}$ base functions, closed under composition, CRN, and B_2RN. Alternatively, it can be defined as those functions containing the L_2 base functions, closed under composition, CRN, and B_2RN.*

Proof. This is proven in Clote and Takeuti [3]. The only difference is that there a slightly different set of initial functions was used. It is well known that all the L_2 base functions are in LOGSPACE. In particular, multiplication is in LOGSPACE. So the algebras above are contained in theirs. On the other hand, the only initial function in their paper that is not an $L_0^{\#}$ base function is BIT, for which an $L_0^{\#}$-term was given above. □

The last definitions needed to present TLS are now given.

Definition 5 *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 $BASIC + open\text{-}LIND$ that t^* is monotonic, and $t \leq t^*$.

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

Definition 6 *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 7 *Ψ-REPL (quantifier replacement) is the following rule:*

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

where $A \in \Psi$.

Definition 8 *TLS is the theory consisting of $BASIC_2$+$open_2$-$LIND$+$\hat{\Sigma}^{\mathsf{b}}_{1,2}$-$WSN$+$\hat{\Sigma}^{\mathsf{b}}_{1,2}$-$REPL$. TLS^- is the same theory but in the language $L_0^{\#}$.*

3 Bootstrapping

This theory is axiomatized in a different fashion than the version of *TLS* presented in Clote-Takeuti [3], so time is needed to show that it is in fact a Σ^{b}_1-conservative extension of their theory. 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. These theories consisted of $BASIC$+$open_2$-$LIND$ and Σ^{b}_0-replacement for C^0_2 and $BASIC$+$open_2$-$LIND$ and the following rule of inference for Δ^b_1-CR:

Definition 9 *Δ^b_1-CR is the following sequent calculus rule:*

$$\frac{A \to B \qquad B \to A}{\to \exists y \leq 2^{|s|} \forall i < |s|(\mathrm{BIT}(i,y) = 1 \equiv A(i,\vec{a}))}$$

where $A \in \Sigma^{\mathsf{b}}_1$ and $B \in \Pi^{\mathsf{b}}_1$.

By quantifier replacement *TLS* contains C^0_2. Johannsen and Pollett [10] show this theory contains Δ^b_1-*CR* and so the latter is also in *TLS*. A direct argument, however, is given below.

Lemma 1 *(1) TLS and TLS^- prove Δ^b_1-CR. (2) $T := BASIC{+}open_2$-$LIND{+}\hat{\Sigma}^{\mathsf{b}}_1$-$WSN$ proves $\hat{\Delta}^b_1$-CR.*

Proof. The same argument shows (1) and (2). Suppose *TLS* proves $A' \to B'$ and $B' \to A'$ where $A' \in \Sigma^{\mathsf{b}}_1$ and $B' \in \Pi^{\mathsf{b}}_1$. Then by quantifier replacement, pairing to get rid of $\wedge$ and $\vee$, and adding dummy quantifiers if needed, it can be assumed that $A' \in \hat{\Sigma}^{\mathsf{b}}_1$ and $B' \in \hat{\Pi}^{\mathsf{b}}_1$. In which case, the formula

$$((b = 1 \wedge A'(i,\vec{a})) \vee (b = 0 \wedge \neg B'(i,\vec{a}))) \wedge$$
$$((x = 1 \wedge A'(Si,\vec{a})) \vee (x = 0 \wedge \neg B'(Si,\vec{a})))$$

is provably equivalent in *TLS* to some $\hat{\Sigma}^{\mathsf{b}}_1$-formula A. If k is taken to be 1, *TLS* also proves there is a unique x such that $A(i,\vec{a},b,x)$. Further, if $k = 1$ then $\mathrm{BIT}(i,w) = \hat{\beta}_{|k|}(i,w)$ is provable in *TLS*, so the w witnessing the existential one gets from the conclusion of the $\hat{\Sigma}^{\mathsf{b}}_{1,2}$-*WSN* rule applied to A can also be used to satisfy the conclusion of the $\Delta^b_{1,2}$-*CR* rules for A' and B'. □

Lemma 2 *TLS and TLS^- prove Δ^b_1-LIND and TLS and TLS^- prove the bit-extensionality axiom:*

$$|a| = |b| \wedge \forall i < |a|(\mathrm{BIT}(i,a) = \mathrm{BIT}(i,b)) \supset a = b\ .$$

Proof. If A is $\Delta^b_{1,2}$ in *TLS*, then *TLS* proves *LIND* for A since *TLS* proves $\exists y \leq 2^{|s|} \forall i < |s|(\mathrm{BIT}(i,y) = 1 \equiv A(i,\vec{a}))$ and since *TLS* proves *LIND* on i for the formula $\mathrm{BIT}(i,y) = 1$. The second statement is easily proved by *LIND* on x in the following $\hat{\Delta}^b_1$-formula:

$$\forall i < |a|(i \leq x \supset \mathrm{BIT}(i,a) = \mathrm{BIT}(i,b)) \supset \mathrm{LSP}(a,x) = \mathrm{LSP}(b,x)\ .$$

□

The next lemma will be useful to show the *TLS* of this paper is Σ^{b}_1-conservative over Clote and Takeuti's *TLS*.

Lemma 3 *$T' :=$ BASIC+open-LIND+Δ_1^b-WSN proves Σ_1^{b}-WSN.*

Proof. Argue informally in T'. If T' proves $b \leq |k(j,\vec{a})| \rightarrow \exists! x \leq |k| A(j,\vec{a},b,x)$, then T' proves $A(j,\vec{a},b,x)$ is equivalent to

$$\forall j \leq |k|[(j < x \vee x < j) \supset \neg A(j,\vec{a},b,x)]$$

which is equivalent to a Π_1^{b}-formula. So T' has WSN for A. □

Write $CTTLS$ for Clote and Takeuti's version of TLS which will be defined in a moment.

Lemma 4 *(1) TLS contains CTTLS. (2) TLS and TLS^- prove their $\hat{\Sigma}_1^{\mathsf{b}}$-definable functions are closed under CRN and B_2RN, and so, contain* LOGSPACE.

Proof. (1) $CTTLS$ is defined using the idea of *essentially sharply bounded (esb) formulas of a theory T*. This is the smallest class containing the atomic formulas, closed under boolean connectives and sharply bounded quantifications, and such that, if A and B are *esb*-formulas and T proves $\exists! x \leq s(\vec{a}) A(\vec{a},x)$, then $C_\Sigma := \exists x \leq s(A(\vec{a},x) \wedge B(\vec{a},x))$ and $C_\Pi := \forall x \leq s(A(\vec{a},x) \supset B(\vec{a},x))$ are esb-formulas. Noticing that $C_\Sigma \Leftrightarrow C_\Pi$, by induction of the complexity of an esb-formula and using Δ_1^b-CR, it can be shown in TLS that every *esb*-formula is a $\hat{\Delta}_1^b$-predicate in TLS. $CTTLS$ was defined in the language $L_0^\#$ and consisted of $BASIC_2$ restricted to this language, bit extensionality, comprehension for esb formulas, and *esb-WSN*, where their WSN looked slightly different from this paper's:

$$\frac{b \leq |k(j,\vec{a})| \rightarrow \exists! x \leq |k| A(j,\vec{a},b,x)}{\begin{array}{c} s \leq |k| \rightarrow \mathrm{Seq}(w) \wedge \mathrm{right}(w) = |k| \wedge \mathrm{Len}(w) = |t| \wedge \beta(1,w) = s \wedge \\ (\exists w \leq (2k+1)\#(4 \cdot (2t+1)^2)) \forall j < |t| A(j,\vec{a},\beta(j+1,w),\beta(j+2,w)) \end{array}}.$$

Here w for the lower sequent is not just a string of blocks, but coded, using a form of sequence coding. Also, the start value of the first element of w is fixed to s. Except for the differences in WSN, though, Lemma 1 and Lemma 2 show that TLS contains $CTTLS$. In Clote and Takeuti sequences are coded as pairs, the right hand side of pair saying the block size the left hand giving the string of blocks. In the case of the witness to the lower sequent's outer existential quantifier for Clote and Takeuti's WSN, the right hand side is chosen to be $|k|$. So given a witness w to the outer existential of the WSN rule in this paper, if the start value condition is pushed into

A and the pair $\langle w, |k| \rangle$ is made, one gets a witness satisfying their WSN's outer existential. Thus, the WSN of this paper implies Clote and Takeuti's. and, as TLS has $\hat{\Sigma}^{\mathsf{b}}_1$-$WSN$, this completes the proof that the TLS contains $CTTLS$.

(2) TLS and TLS^- can $\hat{\Sigma}^{\mathsf{b}}_1$-define any $L^{\#}_0$-term trivially. The proof that these theories are closed under CRN is the same as the proof given in Theorem 4 of Johannsen and Pollett [10]. The proof that these theories are closed under B_2RN is essentially the same as the proof given in Theorem 5.1 of Clote and Takeuti where k is chosen to be $|k'|$. Note quantifier replacement is being used to show that the formula inside the scope of the outer existential quantifier of the lower sequent in a $\hat{\Sigma}^{\mathsf{b}}_{1,2}$-$WSN$ inference is in fact equivalent to a $\hat{\Sigma}^{\mathsf{b}}_1$-formula. □

It is now possible to give an alternative chracterization of $CTTLS$:

Theorem 2 *(1) $CTTLS$ can be equivalently defined as the theory $T :=$ $BASIC$+$open$-$LIND$+Σ^{b}_1-WSN in the language $L^{\#}_0$. (2) $CTTLS$ proves its Σ^{b}_1-definable functions are closed under CRN and B_2RN and so contain* LOGSPACE.

Proof. (1) From the proof of Lemma 4, it follows that $T \supseteq CTTLS$. For the other direction notice that the projection of the left hand side of the outer existential of Clote and Takeuti's WSN gives a witness to the WSN of this paper. So in view of Lemma 3, it suffices to show that the notions of esb and Δ^b_1 coincide for $CTTLS$. This almost follows directly, though, from Clote and Takeuti's witnessing argument to show that the esb-definable functions of $CTTLS$ are the functions in LOGSPACE. To see this note if A is Δ^b_1 in $CTTLS$, then there are $A^{\Sigma} \in \Sigma^{\mathsf{b}}_1$ and $A^{\Pi} \in \Pi^{\mathsf{b}}_1$ such that $CTTLS$ proves

$$\exists y \leq 1(A^{\Sigma}(x) \wedge y = 0) \vee (A^{\Pi}(x) \wedge y = 1).$$

If the witnessing method of Johanssen and Pollett[10] for handling sharply bounded universal quantifiers in front of existential quantifiers is also used, then by Clote and Takeuti's witnessing argument, y can be witnessed by an esb-definable function f. If A^{esb}_f is the defining formula for f, it follows A is equivalent to $A^{esb}_f(x, 1)$. (2) This is proved the same way as in Clote and Takeuti or in the same way as Lemma 4 (2). □

Remark 1 *Besides the theory TLS, Clote and Takeuti consider theories for* $\mathsf{AC}^0(2)$, $\mathsf{AC}^0(6)$, NC^1. *These theories were denoted $TAC^0(2)$, $TAC^0(6)$ and TNC^0 and were obtained by restricting the $|k|$ in Clote and Takeuti's*

WSN rules to be 1, 2, *or any fixed number respectively. Let* m*-WSN denote the restriction that the* $|k|$ *in the WSN rule must be the number* m*. Let LIOpen* := *BASIC+open-LIND in the language* $L_0^{\#}$*. Then by the same reasoning as above* $TAC^0(2)$*,* $TAC^0(6)$ *and* TNC^0 *could be alternately axiomatized as LIOpen together with* Σ_1^b*-1-WSN,* Σ_1^b*-2-WSN, or* Σ_1^b*-*m*-WSN respectively.*

Since *CTTLS* is formulated in the language without multiplication, to get our Σ_1^b-conservation result, the next theorem needs to be established.

Theorem 3 *(1) TLS is a conservative extension of* TLS^-*. (2) CTTLS in* L_2 *with axioms for '*$\cdot$*', denoted* $CTTLS^*$*, is conservative over CTTLS.*

Proof. Both these results are proved in the same way, so only (1) is sketched. To prove (1), a $\hat{\Delta}_1^b$ in TLS^- predicate for $\mathrm{MULT}(x, y, z)$ is given to represent $x \cdot y = z$. To do this, $\mathrm{MULT}(x, y, z)$ will be defined in TLS^- using B_2RN definitions. First, define $\mathrm{NCol}(i, j, n, x, y) :=$

$$\min(\min(\min(\mathrm{BIT}(i, x), \mathrm{BIT}(j, y)), \mathrm{K}_{\leq}(i + j, n)), \mathrm{K}_{\leq}(n, i + j)).$$

This function will be 1 if and only if $i + j = n$ and $\mathrm{BIT}(i, x) \cdot \mathrm{BIT}(j, y)$ is 1. Next define a function $\mathrm{Sumi}(m, j, n, x, y)$ for $\sum_{i=0}^{m} \mathrm{NCol}(i, j, n, x, y)$ as:

$$\begin{aligned}
\mathrm{Sumi}'(0, j, n, x, y) &= 0 \\
\mathrm{Sumi}'(2m, j, n, x, y) &= \\
\mathrm{Sumi}'(2m + 1, j, n, x, y) &= \mathrm{Sumi}'(m, j, n, x, y) + \mathrm{NCol}(m, j, n, x, y) \\
\mathrm{Sumi}(m, j, n, x, y) &= \mathrm{Sumi}'(\mathrm{MSP}(2^{|x|+|y|}, |x| + |y| \dot{-} m), j, n, x, y).
\end{aligned}$$

In turn, define $\mathrm{Sumij}(m, k, n, x, y)$ for $\sum_{j=0}^{k} \sum_{i=0}^{m} \mathrm{NCol}(i, j, n, x, y)$ as

$$\begin{aligned}
\mathrm{Sumij}'(m, 0, n, x, y) &= 0 \\
\mathrm{Sumij}'(m, 2k, n, x, y) &= \\
\mathrm{Sumij}'(m, 2k + 1, n, x, y) &= \mathrm{Sumij}'(m, k, n, x, y) + \mathrm{Sumi}(m, k, n, x, y) \\
\mathrm{Sumij}(m, k, n, x, y) &= \mathrm{Sumij}'(m, \mathrm{MSP}(2^{|x|+|y|}, |x| + |y| \dot{-} k), n, x, y).
\end{aligned}$$

Let $\mathrm{Sum}(n, x, y) := \mathrm{Sumij}(n, n, n, x, y)$. This defines the sum of the bits of the nth column when $x \cdot y$ is computed in the grade school fashion. To calculate the nth bit of $x \cdot y$, one needs to account for carry-bits. Let

CarrySum(n, x, y) be

$$\begin{aligned}
\text{CarrySum}'(0, x, y) &= 0 \\
\text{CarrySum}'(2n, x, y) &= \\
\text{CarrySum}'(2n+1, x, y) &= \lfloor \frac{1}{2}\text{CarrySum}'(n, x, y) \rfloor + \text{Sum}(n, x, y) \\
\text{CarrySum}(n, x, y) &= \text{CarrySum}'(MSP(2^{|x|+|y|}, |x| + |y| \dot{-} n), x, y).
\end{aligned}$$

Then MULT$'(n, x, y)$, which computes the nth bit of $x \cdot y$, can be defined as mod2(CarrySum(n, x, y)), and by Lemma 4, can be 9 $\hat{\Sigma}^{\mathsf{b}}_1$-defined in TLS^-. Thus, MULT(x, y, z) can be defined as:

$$(\forall i \leq |x| + |y|)(\text{MULT}'(n, x, y) = 1 \Leftrightarrow \text{BIT}(n, z)).$$

This is clearly equivalent to a $\hat{\Delta}^b_1$-formula, given that TLS^- has quantifier replacement and that MULT$'(n, x, y) = 1 \equiv \neg$MULT$'(n, x, y) = 0$. The $BASIC_2$ axioms for '$\cdot$' can be shown using this predicate, using the fact that TLS^- has $\hat{\Delta}^b_1$-$LIND$, and using bit extensionality. Standard techniques, such as the technique of induction on theorems explained in Shoenfield [16], can then be used to show using MULT that TLS is conservative over TLS^-. □

The last goal of this section is to give a witnessing argument to show TLS is Σ^{b}_1-conservative over $CTTLS$ and that the $\hat{\Delta}^b_1$-predicates of TLS are LOGSPACE. A bounding term and witness predicate for Σ^{b}_1-formulas are now defined.

- If $A(\vec{a}) \in \Sigma^{\mathsf{b}}_0$ then $t_A = 0$ and $WIT_A(w, \vec{a}) := A(\vec{a}) \wedge w = 0$.
- If $A(\vec{a})$ is of the form $B \circ C$ where $\circ$ is $\wedge$ or $\vee$ then $t_A := 4 \cdot (2^{2|\max(t_B, t_C)|})$ and

$$WIT_A(w, \vec{a}) := \text{ispair}(w) \wedge (WIT_B((w)_1, \vec{a}) \circ WIT_C((w)_2, \vec{a}))$$

- If $A(\vec{a}) \in \Sigma^{\mathsf{b}}_1 \setminus \Sigma^{\mathsf{b}}_0$ is of the form $\exists x \leq tB(x, \vec{a})$ where $B(x, \vec{a})$, then $t_A := 4 \cdot (2^{2|\max(t, t_B)|})$ and

$$WIT_A(w, \vec{a}) := \text{ispair}(w) \wedge (w)_1 \leq t \wedge WIT_B((w)_2, (w)_1, \vec{a}) .$$

- If $A(\vec{a})$ is of the form $\forall x \leq |s| B(x, \vec{a})$ where $B(x, \vec{a}) \in \Sigma^{\mathsf{b}}_1 \setminus \Sigma^{\mathsf{b}}_0$, then $t_A := \text{bd}(t^*_B(|s|), s)$ and

$$WIT_A(w, \vec{a}) := w \leq t_A \wedge \forall x \leq |s| \, WIT_B(\beta(x, |t_A|, w), x, \vec{a})) .$$

The following lemma is true for this witness predicate:

Lemma 5 *If $A(\vec{a}) \in \Sigma^{\mathsf{b}}_1$, then:*

(1) WIT_A is a Σ^{b}_0-predicates.

(2) $TLS \vdash \exists w \leq t_A(\vec{a})\, WIT_A(w, \vec{a}) \supset A(\vec{a})$.

Proof. Part (a) follows from the definition of witness and since $\hat{\beta}$ and the pairing functions are defined by L_0-terms. Part (b) is easily proved by induction on the complexity of A. □

The witness predicate is extended to a witness predicate on cedents in the natural way [1, 9, 14].

Theorem 4 *Suppose*

$$TLS \vdash \Gamma \rightarrow \Delta$$

where Γ and Δ are cedents of Σ^{b}_1-formulas. Let $\vec{a}$ be the free variables in this sequent. (1) There is a LOGSPACE *function f which is $\hat{\Sigma}^{\mathsf{b}}_1$-defined in TLS such that*

$$TLS \vdash WIT_{\wedge\Gamma}(w, \vec{a}) \rightarrow WIT_{\vee\Delta}(f(w, \vec{a}), \vec{a}).$$

(2) There is a LOGSPACE *function f which is Σ^{b}_1-defined in $CTTLS^*$ such that*

$$CTTLS^* \vdash WIT_{\wedge\Gamma}(w, \vec{a}) \rightarrow WIT_{\vee\Delta}(f(w, \vec{a}), \vec{a}).$$

Proof. Both (1) and (2) are proved in the same way, the only difference is that, in the (1) case, the function definition can be shown to be a $\hat{\Sigma}^{\mathsf{b}}_1$-definition in the theory, and in the (2) case, one has to settle for a Σ^{b}_1-definition. Thus, only (1) is shown. The proof is by induction on the number of sequents in a TLS proof of $\Gamma \rightarrow \Delta$. As was already mentioned, by cut elimination, all the sequents in the proof are Σ^{b}_1. The proof breaks into cases depending on the type of inference used for a given line of the proof. All the cases, except $\hat{\Sigma}^{\mathsf{b}}_1$-$WSN$ can be handled essentially as in Johannsen and Pollett [10], so only this case is shown.
($\hat{\Sigma}^{\mathsf{b}}_1$-$WSN$ case) Suppose the inference is:

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

where $A \in \hat{\Sigma}^{\mathsf{b}}_1$. By the induction hypothesis there is a LOGSPACE function g such that

$$WIT_{b\leq|k(j,\vec{a})|}(w,b,j,\vec{a}) \supset WIT_{\exists x\leq|k|A(j,\vec{a},b,x)}(g(w,b,j,\vec{a}),j,\vec{a},b).$$

Hence, as a witness w to the antecedent must equal 0, TLS proves

$$A(j,\vec{a},b,(g(0,b,j,\vec{a}))_1).$$

Using B_2RN, TLS can $\hat{\Sigma}^{\mathsf{b}}_1$-define

$$\begin{aligned} f'(0,\vec{a}) &= (g(0,0,0,\vec{a}))_1 \\ f'(2n,\vec{a}) = f'(2n+1,\vec{a}) &= (g(0,f'(n,\vec{a}),n,\vec{a}))_1 \\ f(j,\vec{a}) &= f'(MSP(2^{|t|},|t|+1\dot{-}j)). \end{aligned}$$

Given this definition, TLS shows

$$\forall j<|t|A(j,\vec{a},f(j,\vec{a}),f(Sj,\vec{a})).$$

So using CRN, TLS can define the sum $\mathrm{Outer}(\vec{a}) := \sum_{j=0}^{|t|-1} f(j,\vec{a})\cdot 2^{j|k|}$ and prove this witnesses the outermost existential of the lower sequent. To get a witness function for $WIT_{\exists w\leq \mathrm{bd}(|k|,t)\forall j<|t|A}$, however, also requires witnesses for the different values of the existential quantifier of A. Witnesses for these values can be defined using B_2RN:

$$\begin{aligned} h'(0,\vec{a}) &= (g(0,0,0,\vec{a}))_2 \\ h'(2n,\vec{a}) = h'(2n+1,\vec{a}) &= (g(0,f'(n,\vec{a}),n,\vec{a}))_2 \\ h(j,\vec{a}) &= h'(MSP(2^{|t|},|t|+1\dot{-}j)). \end{aligned}$$

Note earlier values of h' are not actually needed in the above definition using B_2RN. Given that g witnesses the upper sequent, and that TLS proves the $(g)_1$ that witnesses the quantifier $\exists x\leq|t|$ is the unique value witnessing this x, it follows TLS proves:

$$\forall j<|t|\, WIT_{\forall j<|t|A}(\langle f(Sj,\vec{a}),h(j,\vec{a})\rangle,j,\vec{a},f(j,\vec{a})).$$

Using CRN, TLS can define the sum $\mathrm{Inner}(\vec{a}) := \sum_{j=0}^{|t|-1} h(j,\vec{a})\cdot 2^{j|t_A|}$. From which TLS proves $\langle \mathrm{Outer}(\vec{a}),\mathrm{Inner}(\vec{a})\rangle$ witnesses $WIT_{\exists w\leq \mathrm{bd}(|k|,2t)\forall j<|t|A}$. □

Corollary 1 *The theory TLS is $\Sigma^{\mathsf{b}}_{1,2}$-conservative over $CTTLS^*$. Hence, TLS is $\Sigma^{\mathsf{b}}_{1,L_0^{\#}}$-conservative over $CTTLS$.*

Proof. Suppose $TLS \vdash A$ a Σ_1^{b}-formula. Then by Theorem 4, $CTTLS^*$ proves $WIT_A(f(w,\vec{a}),\vec{a})$, where f is Σ_1^{b}-define in $CTTLS^*$. Let A_f be the formula for this definition. This means that $CTTLS^*$ proves $A_f(w,\vec{a},y) \rightarrow WIT_A(y,\vec{a})$. Since $CTTLS^*$ proves $\rightarrow \exists y \leq t_A A_f$, an $(\exists \leq: right)$, followed by an $(\exists \leq: left)$, followed by a cut, allows the sequent $\rightarrow \exists w' \leq t_A WIT_A(w',\vec{a})$ to be derived. So $CTTLS^*$ proves A by Lemma 5. If A was an $L_0^{\#}$ formula, then $CTTLS$ would prove A by Theorem 3. □

Corollary 2 *(1) The $\hat{\Sigma}_1^{\mathsf{b}}$-definable functions of TLS and TLS^- are exactly* LOGSPACE*. (2) The $\hat{\Delta}_1^b$-predicates of TLS and TLS^- are exactly the* LOGSPACE *predicates. (3) The Σ_1^{b}–definable functions of $CTTLS^*$ and $CTTLS$ are exactly* LOGSPACE*. (4) The Δ_1^b-predicates of $CTTLS^*$ and $CTTLS$ are exactly the* LOGSPACE *predicates.*

Proof. (1) TLS defines all the $\hat{\Sigma}_1^{\mathsf{b}}$-functions by Lemma 4. Suppose $TLS \vdash \forall x \exists! y A(x,y)$. Then by a Parikh's Theorem $TLS \vdash \exists y \leq tA(a,y)$. Taking Γ to be empty in the previous theorem gives a LOGSPACE function $f(a)$ such that $TLS \vdash WIT_A(f(a),a)$. So $TLS \vdash A(a,(f(a))_1)$. The TLS^- result follows from Theorem 3. (2) Suppose f is a predicate in LOGSPACE. Then by Lemma 4, TLS proves $\forall x \exists! y \leq 1 A_f(x,y)$ where A_f is some $\hat{\Sigma}_1^{\mathsf{b}}$-formula for the graph of f. Then TLS proves $A_f(x,1) \Leftrightarrow \neg A_f(x,0)$ and so f predicate is a $\hat{\Delta}_1^b$-predicate. For the other direction, suppose A is $\hat{\Delta}_1^b$ in TLS. Let $A_\Sigma \in \hat{\Sigma}_1^{\mathsf{b}}$ and $A_\Pi \in \hat{\Pi}_1^{\mathsf{b}}$ be equivalent to A. Consider $B(x,y) :=$

$$(\neg A_\Pi(x) \wedge y = 0) \vee (A_\Sigma(x) \wedge y = 1).$$

Certainly, TLS proves $(\forall x)(\exists! y \leq 1) B(x,y)$. The preceding theorem can now be used as in the proof of (1) to get a LOGSPACE predicate.

(3) and (4). Notice TLS, using quantifier replacement, can prove any Σ_1^{b}-formula equivalent to a $\hat{\Sigma}_1^{\mathsf{b}}$. Thus, (3) and (4) follow from Theorem 3 and Corollary 1. □

4 Independence results

To begin we recall some well known results:

Theorem 5 *(Wrathall [19], Kent-Hodgson [8]) (1) The predicates in $\cup_i \hat{\Sigma}_{i,1}^{\mathsf{b}}$ are precisely* LinH*. (2) For $i > 0$, $\hat{\Sigma}_{i,2}^{\mathsf{b}} = \Sigma_i^{\mathsf{p}}$.*

Theorem 6 *(Nepomnjaščiĭ [12])* LinH *contains* TISP$(n^k, n^{1-\epsilon})$*. So* LinH *contains* 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 6 *There is $\hat{\Sigma}^{\mathsf{b}}_{i,1}$-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 $\mathrm{K}_{\leq}$, $\mathrm{K}_{\vee}$, and $\mathrm{K}_{\neg}$, 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 we use 0 as $\ulcorner NOP \urcorner$ meaning no operation. 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

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

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 $e = \langle\langle \ulcorner t_1 \urcorner, \ldots, \ulcorner t_{i+3} \urcorner \rangle\rangle$. This amounts to checking conditions for each m

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

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

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

$$\cdots$$

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

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

Proof. Both results are proved the same way. If A is in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$ then the last argument from Lemma 6 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 8 *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 6 and Theorem 5

$\mathsf{LOGSPACE}$ is contained in LinH, and we have that $\hat{\Sigma}^{\mathsf{b}}_{i,1} = \mathsf{LinH} = \mathsf{PH}$. But by Lemma 7, there are languages in $\hat{\Pi}^{\mathsf{b}}_{i,2}$ that are not in $\hat{\Sigma}^{\mathsf{b}}_{i,1}$. □

Lemma 8 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 second-order class of formulas defining sets similar to $\Sigma^{\mathsf{b}}_{s,1}$. Ferreira's argument was model theoretic. One consequence of Lemma 8 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 9 *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 7 *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 8. □

The next theorem gives another application of Lemma 8.

Theorem 8 *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 8. 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 6. 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., NP = co-NP. As the $\hat{\Delta}^{b}_{1}$-formulas of *TLS* are LOGSPACE, one also gets that $\hat{\Sigma}^{\mathsf{b}}_{1,2} =$ LOGSPACE. But this contradicts Lemma 8. □

Remark 2 *The results presented above are reasonably insensitive to the underlying language as long as the functions symbols added are* 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 6 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 7, 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 $\mathsf{LOGSPACE} \neq \mathsf{NLIN}$. 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. Structure and definability in general bounded arithmetic theories. *Annals of Pure and Applied Logic*. Vol. 100. pages 189–245, October 1999.

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

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

[16] J.R. Shoenfield. *Mathematical Logic*. A.K. Peters, 2001.

[17] 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.

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

[19] C. Wrathall. Complete sets and the polynomial time hierarchy. *Theoretical Computer Science*, 3:23–33, 1976.

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