On the Bounded Version of Hilbert's Tenth Problem

Chris Pollett

214 MacQuarrie Hall
Dept. of Math and Computer Science
San Jose State University
1 Washington Square, San Jose CA 95192
USA

Abstract. The paper establishes lower bounds on the provability of $\mathcal{D}$ =NP and the MRDP theorem in weak fragments of arithmetic. The theory I^5E_1 is shown to be unable to prove $\mathcal{D}$ =NP. This non-provability result is used to show that I^5E_1 cannot prove the MRDP theorem. On the other hand it is shown that I^1E_1 proves $\mathcal{D}$ contains all predicates of the form $(\forall \boldsymbol{i} \leq |b|)P(\boldsymbol{i}, \boldsymbol{x}) \circ Q(\boldsymbol{i}, \boldsymbol{x})$ where $\circ$ is $=$, $<$, or $\leq$, and I^0E_1 proves $\mathcal{D}$ contains all predicates of the form $(\forall \boldsymbol{i} \leq b)P(\boldsymbol{i}, \boldsymbol{x}) = Q(\boldsymbol{i}, \boldsymbol{x})$. Here P and Q are polynomials. A conjecture is made that $\mathcal{D}$ contains NLOGTIME. However, it is shown that this conjecture would not be sufficient to imply $\mathcal{D} = NP$. Weak reductions to equality are then considered as a way of showing $\mathcal{D}$ =NP. It is shown that the bit-wise less than predicate, $\leq_2$, and equality are both co-NLOGTIME complete under FDLOGTIME reductions. This is used to show that if the FDLOGTIME functions are definable in $\mathcal{D}$ then $\mathcal{D} = NP$.

1 Introduction

The Matiyasevich-Robinson-Davis-Putnam (MRDP) theorem [15] says that the Diophantine sets are precisely the recursively enumerable sets and provides a negative answer to Hilbert's Tenth Problem concerning the decidability of the Diophantine sets. The MRDP theorem has been used to show many problems from calculus and differential equations are undecidable (see Matiyasevich [16]).

The study of what bounded versions of this theorem hold was begun by Adleman and Manders [1]. They show a bounded form of MRDP Theorem which says a set A is in $\mathcal{E}_n$ for $n \geq 3$ of the Grzegorczyk Hierarchy iff it is of the form:

$$A = \{x | (\exists \boldsymbol{y} \leq f(x))P(x, \boldsymbol{y}) = Q(x, \boldsymbol{y})\},$$

where P, Q are polynomials with coefficients in $\mathbb{N}$ and f is in $\mathcal{E}_n$. These bounds are quite large as $\mathcal{E}_3$ is already the class of elementary functions. Adleman and Manders also pose the question of whether NP is equal to the class $\mathcal{D}$ of predicates given by formulas of the form:

$$(\exists \boldsymbol{y})[(\sum_j y_j \leq 2^{|\sum_i x_i|^k}) \wedge P(\boldsymbol{x}, \boldsymbol{y}) = Q(\boldsymbol{x}, \boldsymbol{y})]$$

where P, Q are polynomials with coefficents in $\mathbb{N}$. (It should be noted that many papers on this class work with an equivalent formulation where integer coefficients are used and P and Q are pulled-over to the same side of the equality and a check is made if this resulting polynomial is zero. It will be convenient however to work with natural number coefficients in this paper.) They show that the class resulting from the use of only a single existential quantifier is contained in P and moreover if the two variable case is in P then primality checking is p-time. Adleman and Manders [2] define a notion of $\mathcal{D}$-reducibility and give four very weak languages which are $\mathcal{D}$-complete for NP. Jones and Matiyasevich [8] have shown that $\mathcal{D}$ =NP if the predicate $x \leq_2 y$ is in $\mathcal{D}$. Here $x \leq_2 y$ if the i bit of x is less than the i bit of y for all i. Venkatesan and Rajagopalan [21] conjecture that given a number A and a product π of powers of distinct odd primes P_i then testing whether A has even residues modulo all the P_i's is in $\mathcal{D}$. Under this conjecture, they show that $\mathcal{D} = NP$ and that the *Randomized Diophantine Problem* (RDP) is average case complete for NP. On the other hand, $\mathcal{D}$ =NP implies their conjecture, so it follows that $\mathcal{D}$ =NP implies the average case completeness of RDP. Given the scarcity of natural average case complete problems for NP, this lends impetus to resolving the $\mathcal{D} = NP$ problem.

The question of what bounded forms of the MRDP theorem hold is closely connected with how strong a formal system is needed to prove the MRDP theorem. Gaifman and Dimitracopoulos [6] have shown that $I\Delta_0{+}exp$ can prove the MRDP theorem. $I\Delta_0{+}exp$ essentially allows one usual induction on bounded formulas in the language of arithmetic together with the axiom that exponentiation is total. This result was improved by Kaye [11] who showed $IE_1{+}E$ could prove the MRDP theorem and used this to show $IE_1{+}E = I\Delta_0{+}exp$. Here E was an axiom that asserted that a certain Diophantine equation had solutions. The bounds on these solutions are known to be exponential in the real world. These results correspond well with the Adleman and Manders result since Clote and Takeuti [5] have shown that in some sense $I\Delta_0{+}exp$ is the "right" theory to reason about about the elementary functions.

Just as it is unknown if one can get a more than exponentially bounded form of the MRDP theorem, it is also unknown what weaker theories can prove this theorem. It is known that if a theory $T \subseteq S_2$ proves the MRDP theorem then NP=co-NP (see Hájek and Pudlák [7] for a proof). Here S_2 corresponds roughly to a theory which has induction on formulas in the polynomial hierarchy. However, no such complexity consequences are known concerning the provability of $\mathcal{D} = NP$ in subsystems of S_2. It is also still open whether S_2 or a subsystem might be able to prove the MRDP theorem. That is, no one has proven S_2 cannot prove the MRDP theorem. It is known $IOpen$ in the language of $0, S, +, \cdot$ cannot prove MRDP [12].

The first result of this paper is to give a "reasonably strong" theory which cannot prove $\mathcal{D} = NP$. Essentially by results of Kent-Hodgson [14], every NP-predicate can be expressed in the language L_2 mentioned in the abstract by a $\hat{\Sigma}^b_1$-formula. That is, it can be expressed by a formula of the form $B := (\exists y \leq t(\boldsymbol{x}))(\forall i \leq |s(\boldsymbol{x})|)A(x, y, i)$ where A is open (see Pollett [17]). Since $\mathcal{D}$ is closed

under bounded existentiation, to show $\mathcal{D} = NP$ it suffices to show $\mathcal{D}$ contains the class of formulas $\hat{\Pi}_0^b$ of the form $(\forall i \leq |s(\boldsymbol{x})|)A(x, y, i)$. It seems likely that such a proof should be formalizable in a weak arithmetic using induction over values $0 \leq i \leq |s|$ for some $\hat{\Pi}_0^b$-formula and for some $\mathcal{D}$-formula. The theory I^1E_1 which has length induction on bounded existential (E_1) formulas should thus be able to formalize this proof. This is because by standard tricks I^1E_1 can prove length induction on boolean combination of E_1-formulas (so also bounded universal (U_1-) formulas). As evidence for this belief, in this paper it is shown that I^1E_1 proves $\mathcal{D}$ contains the predicates $(\forall \boldsymbol{i} \leq |b|)P(\boldsymbol{i}, \boldsymbol{x}) \circ Q(\boldsymbol{i}, \boldsymbol{x})$ where $\circ$ is $=$, $\leq$, or $<$ and P,Q are polynomials and $\boldsymbol{i} \leq |b|$ means each $i_j \leq |b|$, j=1,..., m. Replacing the L_2-base functions of the open predicate A in the formula B above to get a matrix of the form $P = Q$ for P,Q polynomials would in general introduce bounded existential quantifiers inside the length-bounded universal. So this result does not imply $\mathcal{D}$ =NP. In the case of $=$, one can formalize an old result of Raphael Robinson [19] to show I^0E_1 proves $\mathcal{D}$ contains all predicates of the form $(\forall \boldsymbol{i} \leq b)P(\boldsymbol{i}, \boldsymbol{x}) = Q(\boldsymbol{i}, \boldsymbol{x})$. A "reasonably strong" theory here means I^5E_1, which has five-lengths induction on E_1-predicates and cannot prove $\mathcal{D}$ =NP. The way this is proved is to show that I^5E_1 can $\hat{\Sigma}_1^b$-define the function $\lfloor \frac{1}{3}|x| \rfloor$ but it cannot E_1-define it. Since $\mathcal{D} \subseteq E_1$ this gives the result. The $\hat{\Sigma}_1^b$-definability argument in I^5E_1 essentially follows from a use of excluded middle. The E_1-nondefinability argument is based on a new more sensitive variant of the block counting argument of Johannsen [9, 10] and Pollett [18].

One can slightly sharpen the $\hat{\Sigma}_1^b$-characterization of NP given above. NP can be represented as those formulas of the form $(\exists y \leq t(\boldsymbol{x}))$co-NLOGTIME (The PCP Theorem [3] says something stronger still: that NP is the predicates of the form $(\exists y \leq t(\boldsymbol{x}))$co-RLOGTIME where the co-RLOGTIME machine can only query y constantly many times on a given branch). This is sharper since neither addition nor multiplication is in co-NLOGTIME. It also illustrates how weak a class needs to be shown in $\mathcal{D}$ in order to prove $\mathcal{D}$ =NP and thus illustrates how surprising it is that one can get any reasonable independence proof. This is because this result implies it suffices to show co-NLOGTIME$\subseteq \mathcal{D}$ to get $\mathcal{D} = NP$. Nevertheless, at present the co-NLOGTIME$\subseteq \mathcal{D}$ inclusion seems hard to show; however, the author conjectures that it is reasonably likely that NLOGTIME is in $\mathcal{D}$. This is because the known techniques (machine-based, Chinese Remaindering, or generalized geometric progressions) for removing a bounded universal in front of a block of existentials make two nested uses of exponentiation. The first use is to create a sequence of values and the second nested use is a call usually to $\binom{n}{k}$ to verify some correctness property of this sequence. It seems possible using these techniques that an NLOGTIME computation could be verified by a quantifier of the form $(\forall i \leq ||t||)$ in front of a block of small existentials and polynomial equalities. If two nested exponentials are actually required to do sequence coding with Diophantine equations than this might be the largest machine class one could hope to directly get in $\mathcal{D}$. If NLOGTIME$\subseteq \mathcal{D}$, though, one might conjecture that $\mathcal{D}$ =NP because ENLOGTIME=NP. Here E$\mathcal{C}$ denotes the class of languages expressed by predicates of the form $(\exists y \leq s)(\langle x, y \rangle \in L)$ for some

s an L_2-term and L in $\mathcal{C}$. It is shown however that ENPOLYLOGTIME$\subsetneq$NP. Thus, if $\mathcal{D}$=NP, the proof will need to rely more on the fact that $=$, $+$, and $\cdot$ lie outside of $NLOGTIME$.

Pursuing this last idea, the possibility that $=$ might be hard for co-NLOGTIME under reductions computable in $\mathcal{D}$ is investigated. A function is said to be in FDLOGTIME if it has a polynomial sized bit-graph whose bits are in DLOGTIME. Since $DLOGTIME$ is likely to be in $\mathcal{D}$, it is conjectured that the FDLOGTIME functions are definable in $\mathcal{D}$. It is shown in this paper that both the $\leq_2$ of Jones and Matiyasevich and equality are co-NLOGTIME-complete under FDLOGTIME reductions. So if the conjecture holds then using equality one can show co-NLOGTIME$\subseteq \mathcal{D}$ and so $\mathcal{D}$ =NP. Since it is easy to see that $\mathcal{D}$ =NP implies $\leq_2$ in $\mathcal{D}$, Venkatesan and Rajagopalan [21]'s conjecture, $\leq_2$ in $\mathcal{D}$, the conjecture above, and $\mathcal{D}$ =NP are all equivalently hard problems. In the conclusion we discuss a stronger conjecture which implies but is not equivalent to $\mathcal{D}$ =NP.

The remainder of this paper is organized as follows: In Section 2 some preliminary definitions are presented. In Section 3, it is proven that $EBASIC$ can define $\lfloor \frac{1}{3}|x| \rfloor$ and the function algebras A^m are defined and these algebras are shown to contain all the E_1-definable functions of $I^m E_1$. Then in Section 4, it is established that $\lfloor \frac{1}{3}|x| \rfloor$ is not in A^5 and so $I^5 E_1$ does not prove $\mathcal{D}$ =NP. Section 5 gives the proof that the $(\forall i \leq |s|)P = Q$ predicates are in $\mathcal{D}$. In Section 6 has the results about EPOLYLOGTIME and co-NLOGTIME. Finally, the last section contains a conclusion.

2 Preliminaries

The language L_2 contains the non-logical symbols: 0, S, $+$, $\cdot$, $=$, $\leq$, $\dot{-}$, $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $MSP(x,i)$ and $\#$. 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. $MSP(x,i)$ stands for 'most significant part' and is intended to mean $\lfloor x/2^i \rfloor$. Finally, $x\#y$ 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. The class of quantifier-free formulas is denoted by *open*. For $i > 0$, a $\hat{\Sigma}^b_i$-formula (resp. $\hat{\Pi}^b_i$-formula) is defined to be a E_{i+1}-formula (resp. U_{i+1}-formula) whose innermost quantifier is sharply bounded. Kent and Hodgson [14] (see also Pollett [17]) have shown the sets defined by $\hat{\Sigma}^b_i$-(resp.

$\hat{\Pi}^b_i$-)formulas in the lnaguage of L_2 are precisely the Σ^p_i-(resp. Π^p_i-)predicates. Thus, the $\hat{\Sigma}^b_1$-formulas correspond to the NP predicates.

The theory $BASIC$ is axiomatized by all subtitution instances of a finite set of quantifier free axioms for the non-logical symbols of L_2. These are listed in Buss [4] except for the axioms for MSP and $\dot{-}$ which are listed in Takeuti [20].

The L_2-terms below will be useful for this paper:

$$\begin{aligned}
2^{|y|} = 2^{|y|^1} &:= 1\#y \\
2^{|y|^n} = 2^{1\cdot|y|^n} &:= 2^{|y|^{n-1}}\#y \\
2^{k\cdot|y|^n} &:= 2^{|y|^n}\cdot 2^{(k-1)\cdot|y|^n} \\
2^{\min(|y|,x)} &:= MSP(2^{|y|}, |y| \dot{-} x) \\
K_\neg(x) &:= 1 \dot{-} x \\
K_\wedge(x,y) &:= x \cdot y \\
K_\leq(x,y) &:= K_\neg(y \dot{-} x) \\
LSP(x,i) &:= x \dot{-} MSP(x,i)\cdot 2^{\min(|x|,i)} \\
\hat{\beta}(x,|t|,w) &:= MSP(LSP(w, Sx|t|), x|t|) \\
Bit(i,x) &:= \hat{\beta}(i,1,x) \\
\dot{\beta}(x,|t|,s,w) &:= min(\hat{\beta}(x,|t|,w), s) \\
cond(x,y,z) &:= K_\neg(x)\cdot y + K_\neg(K_\neg(x))\cdot z \\
\max(x,y) &:= cond(K_\leq(x,y), y, x) \\
\min(x,y) &:= cond(K_\leq(x,y), x, y)
\end{aligned}$$

The k and n in $2^{k\cdot|y|^n}$ are fixed integers. Taking products of terms $2^{k\cdot|s|^n}$ one can construct terms representing $2^{p(|s|)}$ where p is any polynomial. Roughly, $\hat{\beta}(x,|t|,w)$ projects out the xth block (starting with a 0th block) of $|t|$ bits from w. $\dot{\beta}(x,|t|,s,w)$ returns the minimum of $\hat{\beta}(x,|t|,w)$ and s. 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 $|t(x)|$ for some $t \in L_2$.

A pairing operation which will sometimes be more convenient than block coding can be defined as follows. Let $B = 2^{|\max(x,y)|+1}$. 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 $\beta(1,w) := \hat{\beta}(0, \lfloor\frac{1}{2}|w|\rfloor \dot{-} 1, \hat{\beta}(0, \lfloor\frac{1}{2}|w|\rfloor, w))$ and $\beta(2,w) := \hat{\beta}(0, \lfloor\frac{1}{2}|w|\rfloor \dot{-} 1, \hat{\beta}(1, \lfloor\frac{1}{2}|w|\rfloor, w))$ which return the left and right coordinates of the pair w. (The real Gödel beta function projects out $\beta(i,w)$, the ith element of a sequence w. However, as this function is never used in this paper, the suggestive notation should not cause confusion.) To check if w is a pair the function $ispair(w) :=$

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

is used. Notice the above functions are all L_2-terms.

Definition 21 *EBASIC is the theory obtained from BASIC by adding the following axioms:*

(1) $b < 2^{\min(k\cdot|d|,|d|^2)} \supset MSP(a\cdot 2^{\min(k\cdot|d|,|d|^2)} + b, \min(k\cdot|d|,|d|^2)) = a.$

(2) $(b < 2^{|d|} \wedge a < 2^{|d|}) \supset (\hat{\beta}(0,|d|,a\cdot 2^{|d|}+b) = b \wedge \hat{\beta}(1,|d|,a\cdot 2^{|d|}+b) = a).$

(3) $Si \cdot |a| \leq k \supset \hat{\beta}(i, |a|, w) = \hat{\beta}(i, |a|, LSP(w, k))$

The new axioms of $EBASIC$ were used in Pollett [17] to show $EBASIC$ can do pairing. In particular, the following lemma was shown.

Lemma 22 *Let* $m = \max(s(a), t(a, s))$, *and let* $t^+ := t(a, \dot{\beta}(0, |m|, s(a), w))$ *where* $s(a), t(a, b) \in L_2$. *Then the theory* $EBASIC$ *proves:*

(a) $(\exists w \leq 2^{2 \cdot |m|}) A(\dot{\beta}(0, |m|, s, w), \dot{\beta}(1, |m|, t^+, w))$
$\Leftrightarrow (\exists x \leq s)(\exists y \leq t) A(x, y)$

(b) $(\forall w \leq 2^{2 \cdot |m|}) A(\dot{\beta}(0, |m|, s, w), \dot{\beta}(1, |m|, t^+, w))$
$\Leftrightarrow (\forall x \leq s)(\forall y \leq t) A(x, y)$.

$BASIC$ and $EBASIC$ proofs will be formalized in the system LKB of Buss [4] where one has equality axioms and where one takes the axioms of $BASIC$ ($EBASIC$) as initial sequents. The main point of LKB is it treats bounded quantifiers syntactically. One defines stronger theories by adding various types of induction rules to $BASIC$ and $EBASIC$.

Definition 23 *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 *is a term in the language, and* $A \in \Psi$. *Here* $|a|_0 = a$ *and* $|a|_m = ||a|_{m-1}|$. *We call* $|t(x)|_m$ *the* principal term *of the induction inference.*

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

Definition 24 $(i \geq 0)$ $I^m E_i$ *is defined to be* $EBASIC$+E_i-$L^m IND$ *and* S_2^i *is defined to be* $BASIC$+$\hat{\Sigma}_i^b$-$LIND$. *Finally,* $S_2 := \cup_i S_2^i$.

3 Definability Results

The proof that $I^5 E_1$ cannot prove $\mathcal{D}$ =NP requires first showing that $I^5 E_1$ can $\hat{\Sigma}_1^b$-define the function $\lfloor \frac{1}{3}|x| \rfloor$ which will be done in this section. Then a function algebra A^m which is an upper bound on the E_1-definable functions of $I^m E_1$ is given. Since the $\hat{\Sigma}_1^b$-predicates are the NP-predicates and the E_1-predicates contain the $\mathcal{D}$-predicates, these two results will serve as the basis for the proof that $I^5 E_1$ cannot prove $\mathcal{D}$ =NP.

Let Ψ be a set of formulas. A theory T can Ψ-*define* a function $f(x)$, if there is a Ψ-formula $A_f(x, y)$ such that $T \vdash \forall x \exists! y A_f(x, y)$ and $\mathbb{N} \models A_f(x, f(x))$. The notions of E_1 and $\hat{\Sigma}_1^b$-definability will be the most useful for this paper.

Theorem 1. $\lfloor \frac{1}{3}|x| \rfloor$ *is* $\hat{\Sigma}_1^b$-*definable in* $EBASIC$.

Proof. Using excluded middle,

$$EBASIC \vdash (\forall x)(\exists y) A_{\lfloor \frac{1}{3}|x| \rfloor}(x, y)$$

where $A_{\lfloor \frac{1}{3}|x| \rfloor}$ is:

$$(\exists z \leq |x|)(B(x,z) \wedge y = z)$$
$$\vee (y = |x| + 1 \wedge \neg(\exists z \leq |x|)B(x,z))$$

and

$$B(x,z) := 3z = |x| \vee 3z + 1 = |x| \vee 3z + 2 = |x|.$$

This is easily seen to be equivalent in $EBASIC$ to a $\hat{\Sigma}_1^b$-formula. Uniqueness follows since $EBASIC$ can prove y must be either less than or equal to $|x|$ or greater than $|x|$ but not both. In the latter case its value is forced to be $|x| + 1$. In the former case, $EBASIC$ can prove only one of $3z = |x|$, $3z + 1 = |x|$, and $3y + 2 = |x|$ can hold by using the axiom $a + b \leq a + c \Leftrightarrow b \leq c$ and equality axioms. It is obvious that $\mathbb{N} \models A_{\lfloor \frac{1}{3}|x| \rfloor}(x, \lfloor \frac{1}{3}|x| \rfloor)$.

Definition 31 *(BPR^m) f is defined from functions g, h, t, and r by m-*length bounded primitive recursion *if*

$$F(0, \boldsymbol{x}) = g(\boldsymbol{x})$$
$$F(n+1, \boldsymbol{x}) = \min(h(n, \boldsymbol{x}, F(n, \boldsymbol{x})), r(n, \boldsymbol{x}))$$
$$f(n, \boldsymbol{x}) = F(|t(n, \boldsymbol{x})|_m, \boldsymbol{x})$$

for some $r \in L_2$ and for some $t \in L_2$.

If g, h, t, and r are multifunctions then f obtained by BPR^m results by viewing each step in the above iteration as a composition of multifunctions.

Definition 32 *A^m is the smallest class containing the L_2-terms, closed under composition, and closed under BPR^m.*

The next theorem gives an upper bound on the E_1-definable functions of $I^m E_1$.

Theorem 2. *The E_1-definable functions of $I^m E_1$ are contained in A^m.*

It is not claimed that $I^m E_1$ can define all the functions in A^m.

Proof. This is proved by a Buss-style [4] "witnessing argument". Let f be E_1-defined by $A_f(\boldsymbol{x}, y)$. If $I^m E_1$ proves $(\forall \boldsymbol{x})(\exists y) A_f$, then since $I^m E_1$ is bounded theory by Parikh's Theorem there is a term $t \in L_2$ such that $I^m E_1$ proves $(\forall \boldsymbol{x})(\exists y \leq t) A_f$. So $I^m E_1$ proves $(\exists y \leq t(\boldsymbol{a})) A_f(\boldsymbol{a}, y)$. By cut-elimination, one can assume every formula of a sequent in this last proof is either of the form $(\exists x_1 \leq t_1)(\exists x_2 \leq t_2)\phi$ where ϕ is an open formula or can be made into this form by padding on dummy quantifiers. Call such a formula LEE_1-formula (lexicographically an existential followed by a E_1-formula). A witness predicate for such an LEE_1-formula is defined as follows: If A is open $Wit_A(w, \boldsymbol{a}) := w = 0 \wedge A(\boldsymbol{a})$. If A is $(\exists x \leq t(\boldsymbol{a}))B$ and A is E_1 then $Wit_A(w, \boldsymbol{a}) := w \leq t(\boldsymbol{a}) \wedge B(w, \boldsymbol{a})$. Finally, if $A(\boldsymbol{a})$ is $(\exists x_1 \leq t_1)(\exists x_2 \leq t_2)B$ and $A \in EE_1$ then

$$Wit_A(w, \boldsymbol{a}) := ispair(w) \wedge \beta(1, w) \leq t_1 \wedge \beta(2, w) \leq t_2 \wedge$$
$$B(\beta(1, 2), \beta(2, w), \boldsymbol{a}).$$

For a cedent $\Gamma = \{A_1, \ldots, A_N\}$ of LEE_1-formulas, let $\wedge\Gamma$ denote their conjunction and $\vee\Gamma$ their negation. Let Δ be another such cedent. Following Pollett [18] one can define open witness predicates $Wit_{\wedge\Gamma}(w, \boldsymbol{a})$, $Wit_{\vee\Delta}(w, \boldsymbol{a})$ for such cedents and also terms t_Γ and t_Δ such that $\mathbb{N} \models \Gamma \to \Delta$ iff

$$\mathbb{N} \models (\exists w \leq t_\Gamma) Wit_{\wedge\Gamma}(w, \boldsymbol{a}) \to (\exists w \leq t_\Delta) Wit_{\vee\Delta}(w, \boldsymbol{a}).$$

The predicate $Wit_{\vee\Delta}(w, \boldsymbol{a})$ in Pollett [18] has the property that if Δ consists of just one formula A then $Wit_{\vee\Delta}(w, \boldsymbol{a}) = Wit_A(w, \boldsymbol{a})$. The predicate $Wit_{\wedge\Gamma}(w, \boldsymbol{a})$ in the case where Γ is empty is defined in Pollett [18] as $0 = 0$.

Next it is argued by induction on proofs of sequents of LEE_1-formulas that if $I^m E_1$ proves $\Gamma \to \Delta$ then there is an A^m function g such that

$$\mathbb{N} \models Wit_{\wedge\Gamma}(w, a) \to Wit_{\vee\Delta}(g(w, a), a).$$

This induction breaks into cases depending on the last inference in the proof of $\Gamma \to \Delta$. All of the cases in this witnessing argument can be handled in the same way as the $i = 1$ case of Theorem 22 in Pollett [18] where $\tau = \{|x|_m\}$. The only difference between the algebra here and the multifunction algebra B_1^τ given there is that the latter contains the $(Wj \leq |t|)(s = 0)$ operator where s, t are L_2-terms. This operator was used only to handle the sharply bounded universal quantifier inferences that arose since $LE\hat{\Sigma}_i^b$-formulas were being considered in that paper. Since all the sequents in our proofs will only involve LEE_1-formulas this kind of inference never arises and so we can get away without these operators.

Thus, if $I^m E_1 \vdash (\exists y \leq t) A_f(a, y)$, then by the above there is a A^m function g such that $\mathbb{N} \models\!\!\to Wit_{A_f}(a, g(a))$. So by the definition of witness, $\mathbb{N} \models A_f(a, \beta(1, g(a)))$. Hence, one obtains $f = \beta(1, g(a)) \in A^m$.

4 $\lfloor \frac{1}{3}|\boldsymbol{x}| \rfloor$ is not $\boldsymbol{E_1}$-definable

In this section, the proof that $I^5 E_1$ cannot prove $\mathcal{D} =$NP is completed by showing that $I^5 E_1$ cannot E_1-define $\lfloor \frac{1}{3}|x| \rfloor$. To show this it suffices by Theorem 2 to show that A^5 cannot define $\lfloor \frac{1}{3}|x| \rfloor$. The idea of the proof is to introduce a notion called the plus-complexity of a natural number. It is shown that if one feeds low plus-complexity inputs into a A^5 function then the plus-complexity of the output of the function will not be too "large". On the other hand, the plus-complexity of $\lfloor \frac{1}{3}|x| \rfloor$ on these inputs will be larger than any A^5 function could produce.

Plus-complexity as defined below will be a more sensitive measure than counting the number of alternations of 1's and 0's in a number that was used in Johannsen [9, 10] and Pollett [18]. Because L_2 contains $|x|$, even if all the inputs to a term t had "few" alternations of 1's and 0's, all one could say about the number of alternations in the output was that it was polynomial in $||\cdots||$ of the inputs. This is not be sufficiently sensitive to show the non-definability of $\lfloor \frac{1}{3}|x| \rfloor$ which is why the more complicated notion of plus-complexity needed to be invented.

Before plus-complexity can be defined, the notion of a *stack code* must first be defined:

Definition 41 *0 is a stack code. If $\bar{v}$ and $\bar{w}$ are stack codes, then the expressions $(\bar{v}+\bar{w})$ and $(2^{\bar{v}})$ and $(-\bar{v})$ are stack codes. Nothing else is a stack code. The natural number v* represented by a stack code $\bar{v}$ *is defined to be the number obtained by evaluating the stack code according to the usual definitions of $+,-$, 2^x on the natural numbers.*

Since 2^0 evaluates to 1 and stack codes are closed under addition it follows that every natural number is represented by at least one stack code. In general, there are many stack codes for any given number. For example, 3 can be represented by $2^{2^0}-2^0$ and by $2^0+2^0+2^0$. Here parentheses are omitted from the codes, as will be done often in this paper, to improve readability. Also, the convention above in the definition that barred variables are stack codes and the same variable without the bar is what it evaluates to will be followed throughout this paper. The *plus-complexity* of a stack code $\bar{v}$, $C_+(\bar{v})$, is defined below.

Definition 42 *If v, w are stack codes then:*

1. $C_+(0) := 1$.
2. $C_+((2^{\bar{v}})) := C_+(\bar{v})$.
3. $C_+((-\bar{v})) := C_+(\bar{v})$.
4. $C_+((\bar{v}+\bar{w})) := C_+(\bar{v}) + C_+(\bar{w}) + 1$.

For a natural number x we define $C_+(x)$ to be

$$\min(C_+(\bar{v}) | \bar{v} \text{ is a stack code evaluating to } x).$$

Lemma 43 *Let $x \in \mathbb{N}$, $x > 0$. (1) There is stack code $\bar{x}$ such that $C_+(\bar{x}) = C_+(x)$ and $\bar{x}$ has the form:*

$$(\cdots(2^{\bar{x}_0} \pm 2^{\bar{x}_1}) \pm \ldots \pm 2^{\bar{x}_n}).$$

where evaluating codes $\bar{x}_i$ to numbers x_i gives $x_0 > x_1 > \ldots > x_n$. (2) If $\bar{x}$ is of the form

$$(\cdots(2^{\bar{x}_0} - 2^{\bar{x}_1})\cdots)$$

then we can take $x_0 \geq x_1 + 2$.

Proof. (1) follows by noticing that stack codes with subcodes of the form $(2^{\bar{y}} + (-2^{\bar{y}}))$ and $(2^{\bar{y}} + 2^{\bar{y}}))$ cannot be minimal. In the minus case, this is because one could replace this subcode with 0. In the plus case, one could replace such a code with $2^{\bar{y}+2^0}$ which has a smaller plus-complexity. To get the $2^{\bar{x}_i}$ ordered in the correct way we can use the fact that $(\bar{v}+\bar{w})$ and $(\bar{w}+\bar{v})$ evaluate to the same number and so do $((\bar{v}+\bar{w})+\bar{z})$ and $(\bar{v}+(\bar{w}+\bar{z}))$ and also $(-(\bar{v}+\bar{w}))$ and $((-\bar{v})+(-\bar{w}))$. Finally, since $x > 0$ the first term does not have a minus in front of it.

For (2) suppose $x_0 = x_1 + 1$ then for the sub-stack code $(2^{\bar{x}_0} - 2^{\bar{x}_1})$ we could have used the code $2^{\bar{x}_1}$ which has lower complexity.

The next lemma gives a simple upper bound on $C_+(x)$.

Lemma 44 *For all $x \in \mathbb{N}$, $C_+(x) \leq |x|^3 + 1$.*

Proof. The proof is by induction on x. $C_+(0) = 1$, $C_+(1) = 1$, $C_+(2) = 1$, $C_+(3) = C_+(2^{2^0} + 2^0) = 3$, $C_+(4) = 1$. So assume for all $n < x$ that $C_+(n) \leq |n|^3 + 1$. Let $\bar{x}$ be the representation of x from Lemma 43. There are fewer than $|x|$ many summands. The plus-complexity rule for plus adds 1 for each of these. By the induction hypothesis, each x_i will have plus-complexity less than $|x_i|^3 + 1 \leq ||x||^3 + 1$. So one can bound the plus-complexity of x by $|x| + |x| \cdot (||x||^3 + 1) = |x| + |x||x||^3 \leq |x|^3 + 1$. The last inequality holds if the weaker inequality $1 + ||x||^3 \leq |x|^2$ holds and this in turn holds if $|x| \geq 3$. i.e, $x \geq 4$.

Definition 45 *The function $\#_B(x)$ returns the number of alternations between 1 and 0 in reading the binary number x from left to right. The counting of this number is started at 1 so $\#_B(1) = 1$.*

As an example, let x be the binary number 1110011 then $\#_B(x) = 3$. The next lemma, gives a lower bound on the plus-complexity of a number.

Lemma 46 *For all $x \in \mathbb{N}$, $C_+(x) \geq \#_B(x) - 1$.*

Proof. Consider a minimal stack code $\bar{x}$ for x. Assume it is written in the form given by Lemma 43. From the definition of plus-complexity in the case of sum, and the fact that each summand has plus-complexity at least 1, we get $C_+(x) \geq 2 \cdot (\text{number of summands} - 1) + 1$ in $\bar{x}$. On the other hand it is easy to see $\#_B(w + 2^y) \leq \#_B(w) + 2$ and also $\#_B(w - 2^y) \leq \#_B(w) + 2$ and so we get the result.

In the next lemma, the plus-complexity of the output of each of the base functions of L_2 is compared with the plus-complexity of its inputs.

Lemma 47 *Let $x, y \in \mathbb{N}$.*

1. $C_+(x + y) \leq C_+(x) + C_+(y) + 1 \leq 2(C_+(x) + C_+(y))$.
2. $C_+(x \dot{-} y) \leq C_+(x) + C_+(y) + 1 \leq 2(C_+(x) + C_+(y))$.
3. $C_+(S(x)) \leq C_+(x) + 2 \leq 3C_+(x)$.
4. $C_+(x \cdot y) \leq (C_+(x) + C_+(y) + 1)^3 \leq 8(C_+(x) + C_+(y))^3$.
5. $C_+(|x|) \leq C_+(x) + 2 \leq 3C_+(x)$.
6. $C_+(x\#y) \leq (C_+(x) + C_+(y) + 5)^3$
   $\leq 4^3(C_+(x) + C_+(y))^3$.
7. $C_+(MSP(x, y)) \leq (C_+(x) + C_+(y) + 1)^2 \leq 4(C_+(x) + C_+(y))^2$.
8. *If $t(x_1, \ldots, x_k)$ is an L_2-term, then one can find fixed constants k, d such that $C_+(t(\boldsymbol{x})) \leq k(\sum_i C_+(x_i))^d$.*

Proof. (1) follows directly from Definition 42.

(2) if $y > x$ then $C_+(x \dot{-} y) = C_+(0) = 1$ and the bound holds; otherwise, the bound holds because of Definition 42 cases (3) and (4).

(3) follows from the fact $C_+(1) = C_+(2^0) = 1$ and the definition of plus-complexity for $+$.

(4) First consider the case where x and y have minimal stack codes of the form $2^{\bar{x}'}$ and $2^{\bar{y}'}$, respectively. Then a stack code for $x \cdot y$ is $2^{\bar{x}'+\bar{y}'}$ and so $C_+(x \cdot y) \leq C_+(x) + C_+(y) + 1$. Notice if there had been minus signs in front of either x or y it would not have affected the plus-complexity of the output. In the general case, a minimal stack code of x (resp. y) will be additions of less than $C_+(x)$ (resp. $C_+(y)$) terms of form $2^{\bar{x}'}$ or $(-2^{\bar{x}'})$ each of plus-complexity less than $C_+(x)$ (resp. $C_+(y)$). So if the minimal stack codes of x and y are "multiplied out" a stack code for $x \cdot y$ is obtained with less than $C_+(x) \cdot C_+(y)$ terms each of which has plus-complexity less than $C_+(x) + C_+(y) + 1$. Using the definition of plus-complexity for addition and subtraction one obtains

$$C_+(x \cdot y) \leq C_+(x)C_+(y)(C_+(x) + C_+(y) + 1) + C_+(x)C_+(y)$$

which is less than

$$(C_+(x)C_+(y) + 1)(C_+(x) + C_+(y) + 1).$$

This in turn is seen to be less than $(C_+(x) + C_+(y) + 1)^3$.

(5) Assume $\bar{x}$ is a minimal stack code for x and is in the form given by Lemma 43. i.e., $\bar{x} = 2^{\bar{x}'} + \bar{v}$ and evaluating $\bar{v}$ gives a number between $-2^{x'-1}$ and $2^{x'}$. Then either $\bar{x}'$, $(\bar{x}' + 2^0)$, or $(\bar{x}' + 2^{2^0})$ is a code for $|x|$. Since $C_+(\bar{x}') \leq C_+(x)$ this gives the bound.

(6) follows from the definition of $x\#y$, (5), (4) and the definitions of $C_+(2^x)$.

(7) Let $\bar{x} := (\cdots(2^{\bar{x}_0} \pm 2^{\bar{x}_1}) \pm \cdots \pm 2^{\bar{x}_n})$ and $\bar{y}$ be stack codes for x and y of minimal complexity as given in Lemma 43. A stack code for $u = x/2^y$ (not necessarily a natural number) is

$$(\cdots(2^{\bar{x}_0 - \bar{y}} \pm 2^{\bar{x}_1 - \bar{y}}) \pm \cdots \pm 2^{\bar{x}_n - \bar{y}}).$$

This code $\bar{u}$ can be split into two parts: a part $\bar{w}$ with $x_i - y \geq 0$ and a part $\bar{v}$ with $x_i - y < 0$. To obtain a stack code for $MSP(x, y) := \lfloor x/2^y \rfloor$, delete $\bar{v}$ from $\bar{u}$ if it evaluates to a positive real and if $\bar{v}$ evaluates to a negative real replace it with $(-(2^0))$. In the resulting stack code there are at most $C_+(x)$ summands of plus-complexity less than $C_+(x) + C_+(y) + 1$. So one can bound the plus-complexity of adding these summands by $(C_+(x) + 1)(C_+(x) + C_+(y) + 1) \leq (C_+(x) + C_+(y) + 1)^2$.

(8) This is straightforward to prove by induction on the complexity of t.

Now the effects of recursion on plus-complexity are considered.

Lemma 48 *Let $C_S(x) := C_+(x) + |||x|||$. Let f be defined by BPR^5 using g, h, r, t satisfying*

1. $C_+(g(\boldsymbol{x})) \leq c \cdot (\sum_i C_S(x_i))^{3^{(|1+\sum_i x_i|_5)^k}}$
2. $C_+(h(\boldsymbol{x}, z, w)) \leq c \cdot (C_S(z) + C_S(w) + \sum_i C_S(x_i))^{\psi}$ *where ψ is defined to be* $3^{(|1+z+w+\sum_i x_i|_5)^k}$.
3. $C_+(r(\boldsymbol{x}, y)) \leq c \cdot (C_S(y) + \sum_i C_S(x_i))^k$ *where $s, s_1 \in L_2$ and c, k are constants.*

Then there is a constant k' such that

$$C_+(f(n,\boldsymbol{x})) \leq c \cdot (C_S(n) + \sum_i C_S(x_i))^{3^{(|1+n+\sum_i x_i|_5)^{k'}}}.$$

Proof. Note there is loss of generality in assuming the constants c, k are the same in the bound of each g, h, and r since if they differed one could always take the maximum of the three values. It follows from the bound on h that

$$C_+(F(n+1,\boldsymbol{x})) \leq c \cdot [C_S(n) + C_S(F(n,\boldsymbol{x})) + \sum_{i=0}^{m} C_S(x_i)]^{\psi}$$

where $\psi := 3^{(|1+n+F(n,\boldsymbol{x})+\sum_i x_i|_5)^k}$. By the definition of BPR^5, $F(n,\boldsymbol{x}) \leq r(n,\boldsymbol{x})$. So also $|F|_3 \leq |r|_3$. Since $r \in L_2$, there is a constant k' such that

$$|r(n,\boldsymbol{x})|_3 \leq k' \cdot (|n|_3 + \sum_{i=0}^{m} |x_i|_3) \leq k' \cdot (C_S(n) + \sum_{i=0}^{m} C_S(x_i))$$

Let $k := k' + 1$. Thus, $C_S(F(n+1,\boldsymbol{x}))$ is less than

$$c[k(C_S(n) + \sum_{i=0}^{m} C_S(x_i)) + C_+(F(n,\boldsymbol{x}))]^{3^{(|1+n+F(n,\boldsymbol{x})+\sum_i x_i|_5)^k}}.$$

Using the bound on $C_+(h)$, one can then expand $C_+(F(n,\boldsymbol{x}))$ and so on. Doing this gives a bound on $C_+(f(n,\boldsymbol{x})) = C_+(F(|t(n,\boldsymbol{x})|_5,\boldsymbol{x}))$ of $Y :=$

$$c \cdot (\sum_{i=0}^{m} C_S(x_i) + \sum_{j=0}^{|t|_5} k \cdot (C_S(j) + \sum_{i=0}^{m} C_S(x_i)))^{3^{\psi}}$$

where ψ is

$$(|1 + \sum_i x_i|_5)^k + \sum_{j=0}^{|t|_5} (W_j)^k$$

and W_j is

$$|1 + j + F(j,\boldsymbol{x}) + \sum_i x_i|_5.$$

The leftmost $\sum_{i=0}^{m} C_S(x_i)$ in Y and the $(|1 + \sum_i x_i|_5)^k$ in ψ come from the composition with g. Since $t \in L_2$ one can bound $|t(n,\boldsymbol{x})|_5 \leq M := (|1 + n + \sum_i x_i|_5)^{k''}$ for some constant k'' and $n \geq 1$. As $F \leq r \leq 2^{|1+n+\sum_i x_i|^d}$ for some d one can bound ψ by $\psi' :=$

$$(|1 + \sum_i x_i|_5)^k + M \cdot (|1 + M + 2^{|1+M+\sum_i x_i|^d} + \sum_i x_i|_5)^k.$$

By choosing d' sufficiently large one can bound ψ' by $\psi'' := (|1+n+\sum_i x_i|_5)^{d'}$. (Notice the term $2^{|1+M+\sum_i x_i|^d}$ is contained within the $|\cdots|_5$.) Now consider the term W under the exponent in Y. Since $|t|_5 \leq M$ and by Lemma 44, W can be bounded by

$$\sum_{i=0}^{m} C_S(x_i) + M \cdot k \cdot (M^3 + 1 + \sum_{i=0}^{m} C_S(x_i)).$$

Notice since $C_+(x_i) \geq 1$ and $M \leq C_S(n) + \sum_i C_S(x_i)$, there are a, b such that the above and W can be bounded by $(W')^b := a \cdot (C_S(n) + \sum_i C_S(x_i))^b$ Thus, $C_+(f) \leq (W')^{b \cdot 3^{\psi''}}$ and one can make this bound into the form of the theorem by choosing a slightly larger value than d' in ψ''.

Corollary 1. *If $f(\boldsymbol{x}) \in A^5$ and $C_+(x_i) \leq |x_i|_3$ then*

$$C_+(f(\boldsymbol{x})) \leq c \cdot (\sum_i |x_i|_3)^{3^{(|\sum x_i|_5)^k}}.$$

Proof. Follows from Lemma 47 and Lemma 48.

The main result of this section can now be established.

Theorem 3. *The function $\lfloor \frac{1}{3}|x| \rfloor$ is not E_1-definable in I^5E_1. Hence, I^5E_1 cannot prove $E_1 = \hat{\Sigma}_1^b$ and also cannot $\mathcal{D} = \hat{\Sigma}_1^b$.*

Proof. By Theorem 2, the E_1-functions of I^5E_1 are contained in A^5. Consider $x := 2^{2^{2^n+1}-1} - 1$ for any n. So for large enough x,

$$C_+(x) \leq (|n|)^3 + 7 \leq |x|_3$$

since by Lemma 44, $C_+(n) \leq (|n|)^3+1$. On the other hand, $\lfloor \frac{1}{3}|x| \rfloor = \lfloor \frac{1}{3}|2^{2^{2^n+1}-1} - 1| \rfloor$ is a number of length $||x|| - 1$ of the form $1010\cdots$. Hence, by Lemma 46,

$$C_+(\lfloor \frac{1}{3}|x| \rfloor) \geq ||x|| - 2 \geq 2^{2^{|x|_4-1}-1} - 2 > c \cdot (|x|_3)^{3^{(|x|_5)^k}}$$

for fixed c, k and large enough x. The last inequality follows because

$$c \cdot (|x|_3)^{3^{(|x|_5)^k}} \leq c \cdot 2^{2^{2 \cdot |x|_5 \cdot (|x|_5)^k}}$$

for fixed c, k, and large enough x. So by Corollary 1, $\lfloor \frac{1}{3}|x| \rfloor$ is not in A^5 and hence not E_1-definable in I^5E_1. But $\lfloor \frac{1}{3}|x| \rfloor$ is $\hat{\Sigma}_1^b$-definable in $EBASIC$. So if I^5E_1 could prove every $\hat{\Sigma}_1^b$-predicate equivalent to some E_1-predicate, it could E_1-define $\lfloor \frac{1}{3}|x| \rfloor$. Thus, I^5E_1 cannot prove $E_1 = \hat{\Sigma}_1^b$. It also cannot prove $\mathcal{D} = \hat{\Sigma}_1^b$ since every $\mathcal{D}$ predicate can be expressed as an E_1-predicate provably in I^5E_1 since even $EBASIC$ can do pairing.

The following is a variation on a result mentioned in the introduction, which is used later to show I^5E_1 cannot prove MRDP.

Theorem 4. *Suppose $T \subseteq S_2$. If T proves the MRDP theorem then T proves $\mathcal{D}$ =NP=PH=co-$\mathcal{D}$.*

Proof. Suppose T proves the MRDP theorem. Then for every U_1-formula $A(\boldsymbol{x})$ there is a formula $E(\boldsymbol{x}) := (\exists \boldsymbol{y})P(\boldsymbol{x},\boldsymbol{y}) = Q(\boldsymbol{x},\boldsymbol{y})$ where P, Q are polynomials such that $T \vdash A \equiv E$. In particular, T proves $A \rightarrow (\exists \boldsymbol{y})P(\boldsymbol{x},\boldsymbol{y}) = Q(\boldsymbol{x},\boldsymbol{y})$. By Parikh's theorem (see Hájek and Pudlák [7] for a proof), since T is a bounded theory one can bound the $\boldsymbol{y}$'s by an L_2-term t. Since A implies this bounded form of E it will also imply $E_2 := (\exists \boldsymbol{y})[|\sum_i y_i| \leq 2^{|\sum_j x_j|^k} \wedge P(\boldsymbol{x},\boldsymbol{y}) = Q(\boldsymbol{x},\boldsymbol{y})]$ for large enough k. Note $E_2 \supset E \supset A$ so $A \equiv E_2$. Hence, every U_1-formula is equivalent to an $\mathcal{D}$-formula. Using this one can show $\mathcal{D} = \Delta_0 = \cup_i E_i = \cup_i \hat{\Sigma}^b_i = PH$. i.e., the polynomial hierarchy (PH) collapses. Since Δ_0 is closed under complement we get $\mathcal{D}$ =co-$\mathcal{D}$.

The next result follows from the previous two theorems.

Corollary 2. *$I^5E_1 \nvdash$ the MRDP theorem.*

In fact, a stronger theory cannot prove the MRDP theorem.

Corollary 3. *Let $Z := \cup_i Z_i$ where $Z_i = EBASIC + \hat{\Sigma}^b_i + L^{i+3}IND$. Then Z cannot prove the MRDP theorem.*

Proof. In Pollett [18], it was shown Z cannot prove the collapse of the polynomial hierarchy, but $\mathcal{D}$ =NP=co-$\mathcal{D}$ implies the collapse of the polynomial hierarchy.

5 Towards $\mathcal{D}$ =NP

In this section, several statements in the direction of trying to show $\mathcal{D}$ =NP will be proven. It will be shown that I^1E_1 proves $\mathcal{D}$ contains predicates of the form $(\forall i \leq |s|)P \circ Q$ for $\circ$ either $=$, $\leq$, or $<$. A $\mathcal{D}$-predicate $A(\boldsymbol{x}) :=$

$$(\exists \boldsymbol{y})[(\sum_j y_j \leq 2^{|\sum_i x_i|^k}) \wedge P = Q]$$

is *u$\mathcal{D}$ in T*, if T proves $(\exists \boldsymbol{y})(P = Q) \supset A$. It is shown below that I^1E_1 proves its u$\mathcal{D}$-functions closed under composition, and under $\wedge$, and $\vee$. This is used to show I^1E_1 can $\mathcal{D}$-define L_2-terms. Lastly, $\mathcal{D}$ is shown to be closed under certain sums of polynomial length and this is used to prove the result.

Lemma 51 *Let P_i, Q_i be polynomials EBASIC proves (1) $(P_1 = Q_1 \wedge P_2 = Q_2) \Leftrightarrow P_1 + P_2 = Q_1 + Q_2$. (2) $(P_1 = Q_1 \vee P_2 = Q_2) \Leftrightarrow P_1P_2 + Q_1Q_2 = P_1Q_2 + P_2Q_1$ (3) I^1E_1 proves its u$\mathcal{D}$-definable functions are closed under composition.*

Proof. (1) and (2) are straightforward. For (3), let f, g be two Diophantine functions. By Parikh's theorem, if I^1E_1 proves $(\forall x)(\exists! y)A_f(x,y)$ then there is an L_2-term t_f bounding y such that $I^1E_1 \vdash (\forall x)(\exists y \leq t_f)A_f(x,y)$. Since y was originally unbounded and unique one can choose t_f of the form $2^{|x|^k}$ for some fixed k. So if and $I^1E_1 \vdash (\forall y)(\exists! z)A_g(y,z)$. Then it follows I^1E_1 proves $(\forall x)(\exists! z)(\exists y \leq 2^{|x|^k})A_f \wedge A_g$. let B be what's inside the scope of the $(\exists! z)$. Then B can be made into a $\mathcal{D}$ predicate by choosing a bound of the correct form larger than the bounds on y and in A_f and A_g, prenexifying the existentials of A_f and A_g and using (1). To show that this new predicate, C, is equivalent to B, we use that A_f and A_g are $u\mathcal{D}$ for I^1E_1 to show $C \supset B$.

Lemma 52 *I^1E_1 can $u\mathcal{D}$-define $<$, $\leq$, $x \dot{-} y$, and $\lfloor x/y \rfloor$ and prove basic properties of them.*

Proof. Define $LE(x,y)$ as $(\exists w)(w \leq 2^{|x+y|} \wedge x + w = y)$ and $x < y$ as $LE(Sx, y)$. The goal is to show I^1E_1 proves $LE(x,y) \Leftrightarrow x \leq y$ where $\leq$ is the L_2-predicate symbol. That $LE(x,y)$ implies $x \leq y$ involves checking that LE satisfies all of the $EBASIC$ axioms for $\leq$. For instance, $LE(0,y)$ holds taking $w = y$. For $x \leq y$ implies $LE(x,y)$, use the $EBASIC$ axioms (4) and (20) from Buss [4] and LIND on these axioms to binary search for the w such that $x + w = y$. Define $Sub(x,y) = z$ as $(x = y + z \vee (z = 0 \wedge x < y))$. Since $\dot{-}$ is in L_2, I^1E_1 can use this to prove z exists and is unique and in fact Takeuti's [20] axiom for $\dot{-}$ is precisely how $Sub(x,y)$ is defined. Define $\lfloor x/y \rfloor = z$ as $(\exists w)[w \leq 2^{|x+y|} \wedge ((y = 0 \wedge z = 0 \wedge w = 0) \vee (w < y \wedge x = zy + w))]$. The proof of existence of z in I^1E_1 can be done using $LIND$ to binary search for a value. Uniqueness and being in $u\mathcal{D}$ follow from $EBASIC$ axioms.

The next argument is based on an observation of Raphael Robinson [19].

Theorem 5. *Let P, Q be polynomials. (1) I^0E_1 proves $\mathcal{D}$ contains the predicate $B_=(b) := (\forall \boldsymbol{i} \leq b)P(\boldsymbol{i}, \boldsymbol{x}) = Q(\boldsymbol{i}, \boldsymbol{x})$. (2) I^1E_1 proves $\mathcal{D}$ contains the predicate $B_=(|b|)$.*

Proof. First, note the predicate $B_=(b)$ above holds if and only if $P'(b, \boldsymbol{x}) := \sum_{i_1=0}^{b} \cdots \sum_{i_k=0}^{b} P(\boldsymbol{i}, \boldsymbol{x})$ is equal to $Q'(b, \boldsymbol{x}) := \sum_{i_1=0}^{b} \cdots \sum_{i_k=0}^{b} Q(\boldsymbol{i}, \boldsymbol{x})$. P' and Q' can be written as polynomials. This is because $\sum_{i_j=0}^{b} (i_j)^m$ for fixed m is expressible as a degree $m+1$ polynomial over a fixed number. For example $\sum_{i_j=0}^{b} (i_j)^1 = b(b+1)/2$. Let $C(c)$ be the formula: $B_=(c) \Leftrightarrow P'(c, \boldsymbol{x}) = Q'(c, \boldsymbol{x})$. As mentioned in the introduction by standard methods [17] one can show I^0E_1 proves induction for Boolean combinations of E_1-formulas. So it has induction for $C(c)$. I^0E_1 trivially shows $C(0)$ holds and I^0E_1 proves $P'(c, \boldsymbol{x}) + \sum_{j_1=0}^{1} \cdots \sum_{j_k=0}^{1} P(c + j_1, \ldots, c + j_k, \boldsymbol{x})$ equals $P'(c+1, \boldsymbol{x})$ and similarly for $Q'(c+1, \boldsymbol{x})$. This suffices to show $C(c+1)$. So by induction on c, I^0E_1 can prove $C(b)$ which shows $B_=$ is in $\mathcal{D}$.

A Diophantine representation of exponentiation was given in Matiyasevich [15] by studying the zeros of the equation $p(a,x,y) := x^2 + y^2 - 2ay - 1$. Adleman

and Manders [1] have shown the predicate $E(x,y,z) := x = y^z$ is in $\mathcal{D}$ by bounding the existentials in this representation in terms of x. Using $E(x,y,z)$ one can define $lh(x) = z$ as $(\exists y)(x \leq y \wedge y \leq 2x \wedge E(y,2,z))$ and one can define $x\#y = z$ as $E(z,2,lh(x)\cdot lh(y))$. (These are not quite $\mathcal{D}$ predicates but one can combine the bound on y in lh with that of E to make it in $\mathcal{D}$.) I^1E_1 can prove the existence of z in $lh(x) = z$ using LIND on this predicate and can also show $lh(x) = |x|$ and show E and lh are $u\mathcal{D}$-predicates. The details about verifying the correctness of the E predicate in weak theories of arithmetic can be found in Kaye [11]. Together with Lemma 52, I^1E_1 can show all the L_2 base functions are in $u\mathcal{D}$ and, hence, also the L_2-terms listed in the preliminaries.

Lemma 53 *Fix n and suppose $|i^n| < |b|$ for all $i \leq |a|$. Then (1) I^1E_1 can $\mathcal{D}$-define $S_n(|a|) := \sum_{i=0}^{|a|} i^n 2^{i\cdot|b|}$ and (2) prove $S_n(|2a|) = S_n(|a|+1) = S_n(|a|) + (|a|+1)^n$.*

Proof. The proof is by induction (in the real world) on n. For $n = 0$ note I^1E_1 can $\mathcal{D}$-define $S_0(|a|)$ using the formula for summing a geometric series: $\lfloor \frac{2\cdot 2^{|b||a|}-1}{2^{|b|}-1} \rfloor$. Now assume for $n' < n+1$ we have defined $S_{n'}(|a|)$ in I^1E_1 and proven (2). Then one can define $S_{n+1}(|a|) = x$ as

$$(2^{|a|}-1)x = |a|^{n+1}2^{|b|} - 1 + \sum_{i=0}^{n}(-1)^{n+1-i}\binom{n+1}{i}(S_i(|2a|) - S_i(|0|)).$$

The value of an x satisfying the above can be bounded by the right hand side of the equation. I^1E_1 can verify (2) for S_{n+1} by LIND on c for the equation for $S_{n+1}(\min(c,|2a|))$, the induction step using that S'_n is correctly defined. The above definition is motivated by the appendix on summing geometric equations in Matiyasevich [16] and by Kaye [13]. Note that n in the above (and so the $\binom{n+1}{i}$'s) are fixed natural numbers.

Lemma 54 *Suppose $|P(i,\boldsymbol{y})| < |s|$ for all $i \leq |t|$. Then I^1E_1 can (1) $\mathcal{D}$-define $S(|a|,\boldsymbol{y}) := \sum_{i=0}^{|a|} P(i,\boldsymbol{y})2^{i\cdot|s|}$ and (2) prove $S(|2a|,\boldsymbol{y}) = S(|a|+1) = S(|a|,\boldsymbol{y}) + P(|a|+1,\boldsymbol{y})$.*

Proof. Rewrite P as $\sum_{j=0}^{d} P_j(\boldsymbol{y})i^j$ where the P_j's are polynomials in the remaining variables. So S equals

$$\sum_{i=0}^{|a|}(\sum_{j=0}^{d} P_j(\boldsymbol{y})i^j)\cdot 2^{i\cdot|s|}$$

which in turn equals

$$\sum_{j=0}^{d} P_j(\boldsymbol{y})(\sum_{i=0}^{|a|} i^j \cdot 2^{i\cdot|s|}).$$

Each of the outer summands is just a product of a polynomial with the kind of sums we showed $\mathcal{D}$-definable in Lemma 53 (1). So S is $\mathcal{D}$-definable in I^1E_1. Similarly using Lemma 53 (2) establishes (2) above.

Theorem 6. *Let P, Q be polynomials. I^1E_1 proves $B_\circ(|b|, \boldsymbol{y}) := (\forall \boldsymbol{i} \leq |b|)P(\boldsymbol{i}, \boldsymbol{y}) \circ Q(\boldsymbol{i}, \boldsymbol{y})$ is equivalent to a $\mathcal{D}$-predicate where $\circ$ is either $<$ or $\leq$.*

Proof. Only the $B_<$ case is shown as the proofs of the two cases are essentially the same. Since P, Q have nonnegative integer coefficients they are nondecreasing in each variable. Hence, I^1E_1 proves $P(\boldsymbol{i}, \boldsymbol{y}) \leq P(|b|, \ldots, |b|, \boldsymbol{y})$ and $Q(\boldsymbol{i}, \boldsymbol{y}) \leq Q(|b|, \ldots, |b|, \boldsymbol{y})$. Let $m := |P(|b|, \ldots |b|, \boldsymbol{y}) \cdot Q(|b|, \ldots |b|, \boldsymbol{y})|$. Now $B_<$ holds iff we have

$$(\sum_{i_1=0}^{|b|} \cdots \sum_{i_k=0}^{|b|} (P(\boldsymbol{i}, \boldsymbol{y}) + 1 + \beta(v, m, w))2^v)$$

where $v = \sum_{j=0}^{k} i_j \cdot m^j$ is equal to

$$(\sum_{i_1=0}^{|b|} \cdots \sum_{i_k=0}^{|b|} Q(i, \boldsymbol{y})2^{\sum_{j=0}^{k} i_j \cdot m^j})$$

for some polynomial length string w. Let $L(|b|, \boldsymbol{y}, w)$ and $R(|b|, \boldsymbol{y})$ be the first and second sums respectively. Using Lemma 54 and that the L_2-terms are $u\mathcal{D}$-definable, I^1E_1 can $\mathcal{D}$ define L and R. So LIND on c in $C(c) :=$

$$B_<(\min(c, |b|), \boldsymbol{y}) \Leftrightarrow (\exists w \leq t)(L(\min(c, |b|), \boldsymbol{y}, w) = R(\min(c, |b|), \boldsymbol{y}))$$

for some large enough term t gives the result. The induction step uses Lemma 54(2). As mentioned in the introduction I^1E_1 will have LIND for this predicate since it is a boolean combination of E_1-formulas.

6 Logtime classes and $\mathcal{D}$

As was mentioned in the introduction the author conjectures that techniques similar to those in the last section might be able to show NLOGTIME is contained in $\mathcal{D}$. In this section, it is shown that this result would not suffice to prove $\mathcal{D}$ =NP. The possibility of proving $\mathcal{D}$ =NP by showing equality is hard for co-NLOGTIME under some kind of reduction computable in $\mathcal{D}$ is then considered. The model of sublinear time used in this section is the standard one: The machines used are allowed to write an index of an input tape square on a query tape, enter a query state, and in one step get the value of that square of the input.

Definition 61 *Let $\mathcal{C}$ be a class of languages. Then $E\mathcal{C}$ denotes the class of languages in $(\exists y \leq s)\langle x, y\rangle \in L$ for some s an L_2-term and L in $\mathcal{C}$.*

One might conjecture that $ELOGTIME$ =NP or the weaker conjecture that ENPOLYLOGTIME=NP. It is established next that this is not possible.

Theorem 7. *Let $t(|x|)$ be monotonic L_2-terms so that $t(|x|) \geq \log |x|$, $t(|x|^k) \in O([t(|x|)]^k) \subset o(|x|)$ for all k. Then*

$$ENTIME[t^{O(1)}] = NTIME[t^{O(1)}].$$

The case of a single input variable x is shown but the method can be generalized to any number of inputs.

Proof. By the definition if L is in ENTIME$[t^{O(1)}]$, there is some L_2-term s and L' in NTIME$[t^{O(1)}]$ such that

$$x \in L \Leftrightarrow (\exists y \leq s)\langle x, y\rangle \in L'.$$

Note since s is an L_2-term then s can be bounded above by $2^{p(|x|)}$ for some polynomial p. Thus, there is some nondeterministic machine M for L' such that on inputs x and $y \leq s(x)$ decides if $x \in L$ in time bounded by $[t(|x| + p(|x|))]^k \leq t(|x|)^{k'}$ for some constants k, k'. Without loss of generality one can assume M never queries the $|s| - 1$st bit of y, so if M makes a query to the i bit of y there are values of $y \leq s$ so that this value is both 1 and 0. Let N be the machine that simulates M on input x but whenever M makes a query of a bit value of y, N checks in a table it stores on another tape if that query has been made before. If it has it uses the value it stored for that query. Otherwise, N nondeterministically guesses 1 or 0 as a returned value. Storing the value of which bit position was queried takes less than $\log |s| \leq [t(|x|)]^m$ bits for some m so N's runtime is in NTIME$[t^{O(1)}]$. If there was a value of y such that M would accept, then the bits used in this y give a guess path to an accepting path in N. Since all possible bit settings for y's occur in values less than s if there were some nondeterministic choices that make N accept, one could set the appropriate bits of y accordingly and make M accept.

Corollary 4. *ENPOLYLOGTIME$\subsetneq$NP.*

Proof. By the Theorem 7

$$\text{ENPOLYLOGTIME} \subseteq \text{NLIN} \subsetneq \text{NP}.$$

The last strict inclusion is by the nondeterministic time hierarchy theorem.

It might be possible to show $\mathcal{D}$ =NP by showing equality is hard for co-NLOGTIME under some kind of reduction. As defined in the introduction, a polynomially bounded functions is in FDLOGTIME if each bit of its output is DLOGTIME computable. If NLOGTIME is in $\mathcal{D}$ then computing any individual bit of an FDLOGTIME functions will be in $\mathcal{D}$; however, it would still be unclear if given a FDLOGTIME function whether its graph would be in $\mathcal{D}$. The author conjectures it will be and the next theorem shows that this conjecture would imply $\mathcal{D}$ =NP. Recall from the introduction that the predicate $x \leq_2 y$ holds if the ith bit of x is less than the ith bit of y for all i.

Theorem 8. *$x \leq_2 y$ and equality are co-NLOGTIME-complete under FDLOGTIME reductions.*

Proof. Both problems are in co-NLOGTIME as one can universally guess a bit position i and then in DLOGTIME check if either $BIT(i, x) \leq BIT(i, y)$ or

$BIT(i,x) = BIT(i,y)$. Let M be a co-NLOGTIME machine and x an input. Without loss of generality one can assume each path in the computation tree of M is of length $k \log |x|$. The FDLOGTIME reduction outputs two numbers w, v to be input to the $\leq_2$ predicate as $w \leq_2 v$. It sets the ith bit of w to 1 if M on path i rejected and to 0 otherwise. It sets the ith bit of v always to 0. Since on a fixed path M's computation is DLOGTIME, this reduction will be in FDLOGTIME. Thus, $w \leq_2 v$ iff for every bit of w is 0 and this happens iff M accepted on every path. This in turn implies $x \in L(M)$. To reduce x, M to equality let the ith bit of w be 1 iff M accepts on path i and let the ith bit of v be 1. Again, this reduction is in FDLOGTIME and $w = v$ iff M accepts on every path iff $x \in L(M)$.

7 Conclusion

Several lines of attack on the $\mathcal{D}$ =NP question seem promising. It is quite likely one can find a better complexity notion than plus-complexity such that for all the L_2 base functions the complexity of the output can be bounded linearly in the outputs. This measure would also have to be useful in that one should be able to use the excluded middle trick to define some function whose complexity cannot easily be bounded this new measure. Nevertheless, such a measure might exist and allow one to show $I^m E_1$ cannot prove $\mathcal{D}$ =NP for some $m < 5$.

Towards showing that $\mathcal{D}$=NP holds, the approach of considering reductions to equality might still bear fruit. The first step would be to show a complexity class like NLOGTIME or DLOGTIME is contained in $\mathcal{D}$. This would not immediately imply FDLOGTIME is contained in $\mathcal{D}$. But maybe some weaker reduction would suffice. In any case, it is useful to make conjectures which aren't already equivalent to $\mathcal{D}$ =NP in the hopes that they either lead one towards a proof of $\mathcal{D}$ =NP or their refutations gives some new insight on the problem. For instance, one reducibility which is presumably weaker than FDLOGTIME reducibility is to consider functions whose graphs are recognizable in DLOGTIME (or NLOGTIME). A reasonable conjecture is that equality testing is hard for co-RLOGTIME under this kind of reducibility. If DLOGTIME is in $\mathcal{D}$, this would suffice to show $\mathcal{D}$ =NP. On the hand, $\mathcal{D}$ =NP does not neccessarily imply this conjecture. Unfortunately, these kinds of reductions are probably too weak to show the desired hardness result.

References

1. L.M. Adleman and K. Manders. The computational complexity of decision procedures for polynomials. In *16th Annual Symposium on Foundations of Computer Science*, pages 169-177. 1975.
2. L.M. Adleman and K. Manders. Diophantine Complexity In *17th Annual Symposium on Foundations of Computer Science*, pages 81-88. 1976.
3. S. Arora, C. Lund, R. Motwani, M.Sudan, and M.Szegedy Proof verification and hardness of approximation problems. In *33th Annual Symposium on Foundations of Computer Science*, pages 14-23. 1992.

4. S.R. Buss. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.
5. P. Clote and G. Takeuti. Exponential time and bounded arithmetic (extended abstract). In *Structure in Complexity Theory LNCS223*, pages 125–143. Springer-Verlag, 1986.
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. J.P. Jones and Y. Matiyasevich. Register machine proof of the theorem on exponential diophantine representation. *Journal of Symbolic Logic*, 49:818–829, 1984.
9. J. Johannsen. On the weakness of sharply bounded polynomial induction. In *Proceedings of Kurt Gödel Colloquium 1993*, pages 223–230. Springer-Verlag, 1993.
10. J. Johannsen. A model-theoretic property of sharply bounded formula with some applications. *Mathematical Logic Quarterly*, 44(2):205–215, 1998.
11. R. Kaye. Diophantine induction. *Annals of Pure and Applied Logic*, 46:1–40, 1990.
12. R. Kaye. Open induction, Tennenbaum phenomena and complexity theory. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 222–237. Oxford Science Publications, 1993.
13. R. Kaye. A diophantine undecidable subsystem of arithmetic with no induction axioms. To appear in Journal of Symbolic Logic.
14. C. F. Kent and B.R. Hodgson. An arithmetical characterization of NP. *Theoretical Computer Science*, 21:255–267, 1982.
15. Y. Matiyasevich. Enumerable sets are Diophantine. *Dokl. Acad. Nauk*, 191:279–282, 1970.
16. Y. Matiyasevich. *Hilbert's Tenth Problem*. MIT press, 1993.
17. C. Pollett. Structure and definability in general bounded arithmetic theories. Annals of Pure and Applied Logic. 100:189–245, October 1999.
18. C. Pollett. Multifunction algebras and the provability of $PH\downarrow$. Annals of Pure and Applied Logic. 104:279–303. July 2000.
19. R. Robinson. Arithmetical representation of recursively enumerable sets. Journal of Symbolic Logic 21(2):162–186. June 1956.
20. G. Takeuti. *RSUV* isomorphisms. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–386. Oxford Science Publications, 1993.
21. R. Venkatesan and S. Rajagopalan. Average case intractibility of matrix and Diophantine problems. In *Proceedings of the 24th Symposium on the Theory of Computation 1992*, pages 632–642. ACM press, 1992.