On Proofs About Threshold Circuits and Counting Hierarchies (Extended Abstract)

Jan Johannsen*

Department of Mathematics
University of California, San Diego
La Jolla, CA 91093-0112

Chris Pollett

Department of Computer Science
Boston University
Boston, MA 02215

Abstract

We define theories of Bounded Arithmetic characterizing classes of functions computable by constant-depth threshold circuits of polynomial and quasipolynomial size. Then we define certain second-order theories and show that they characterize the functions in the Counting Hierarchy. Finally we show that the former theories are isomorphic to the latter via the so-called RSUV-isomorphism.

1 Introduction

A phenomenon that is commonly observed in Complexity Theory is that proofs of results about counting complexity classes ($\#P$, Mod_pP etc.) can often be scaled down to yield results about small depth circuit classes with the corresponding counting gates. For example, Toda's result [17] that every problem in the Polynomial Hierarchy can be solved in polynomial time with an oracle for $\#P$ corresponds to Allender's theorem [1] that polynomial size constant-depth circuits with unbounded fan-in AND and OR gates can be simulated by quasi-polynomial size depth 3 threshold circuits.

We give a logical explanation for this phenomenon and turn the observation into a theorem by defining bounded arithmetic theories corresponding to the Counting Hierarchy FCH (which is the union of $\#P$, $\#P^{\#P}$, $\#P^{\#P^{\#P}}$ ... and can be viewed as the largest counting class) on the one hand and constant-depth threshold circuits (TC^0-circuits) of quasi-polynomial size on the other hand, and showing that they are isomorphic.

The paper is organized as follows: First we give characterizations of the classes of functions computable by constant-depth threshold circuits of polynomial and quasi-polynomial size, and more generally of size $exp(exp((\log\log n)^{O(1)}))$, $exp(exp(exp((\log\log\log n)^{O(1)})))$ ... , by function algebras. In order to do that, we give a new proof of Clote and Takeuti's [9] function algebra characterization of the functions computed by polynomial size TC^0 circuits. Unlike the original proof, ours can be generalized to the case of quasi-polynomial and the above larger size bounds.

We then define a hierarchy of bounded arithmetic theories C^0_k for $k \geq 2$, and show that these theories characterize the above classes of threshold circuits. More precisely, the functions whose graphs are defined by bounded existential formulas and that are provably total in the theories C^0_2 and C^0_3 are precisely those in computable by polynomial size and quasipolynomial size TC^0-circuits, and analogous relations hold between the theories C^0_k for $k > 3$ and the larger size threshold circuit classes mentioned above. This simplifies and generalizes earlier work by the first author [10].

Next we define another hierarchy of second-order bounded arithmetic theories $\mathbf{D}^0_k$ for $k \geq 1$. Using the function algebra characterization of the counting hierarchy FCH by Vollmer and Wagner [18], we then show that the theory $\mathbf{D}^0_2$ characterizes FCH: The functions provably total in $\mathbf{D}^0_2$ whose graphs are definable by second-order existential bounded formulas are exactly the functions in FCH. Similarly, the theories $\mathbf{D}^0_k$ with $k > 2$ correspond to classes defined analogous to FCH, but using machines with quasi-polynomial (for $k = 3$) and longer running times. The witnessing argument that we use to prove these results is simpler than the second-order witnessing of Buss [3] and could also be applied to give simpler proofs of earlier results concerning second-order bounded arithmetic.

Finally we show that for every $k \geq 1$, the theories C^0_{k+1} and $\mathbf{D}^0_k$ are isomorphic via the so-called $RSUV$-isomorphism [14, 16]. The idea behind this isomorphism is that a number a can be viewed as the set

* Supported by DFG grant No. Jo 291/1-1

$\{ i \,;\, \text{the } i\text{th bit in } a \text{ is } 1 \}$, and vice versa a finite set A can be viewed as representing the number $\sum_{a\in A} 2^a$. This way, the numbers in a second-order theory correspond to the *small* numbers in a first-order theory, i.e. those in the range of the logarithm function, whereas the sets in a second-order theory correspond to all numbers in a first-order theory.

Technically, this means that there are translations mapping a first-order formula A to a second-order formula A^H, and a second-order formula B to a first-order formula B^L such that $\mathbf{D}^0_k$ proves A^H for every theorem A of C^0_{k+1}, and C^0_{k+1} proves B^L for every theorem B of $\mathbf{D}^0_k$. These statements are proved by induction on the lengths of proofs, so that a proof in one of the theories can be translated step by step into a proof in the other theory. Moreover A and A^{HL} are provably equivalent in C^0_{k+1}, and for a bounded formula B, B and B^{LH} are provably equivalent in $\mathbf{D}^0_k$, so the translations indeed form a kind of isomorphism between theories.

2 Function Algebras

We define a hierarchy of growth rates by $\tau_1(n) := O(n)$, and then inductively $\tau_{k+1}(n) := 2^{\tau_k(\log n)}$. In particular, $\tau_2(n)$ are the polynomial and $\tau_3(n)$ are the quasi-polynomial growth functions.

Let $TC^0(f(n))$ denote the set of functions computable by Dlogtime-uniform families of threshold circuits of constant depth and size $O(f(n))$, and let TC^0 abbreviate $TC^0(\tau_2(n)) = TC^0(n^{O(1)})$ and $qTC^0 = TC^0(\tau_3(n))$. Thus TC^0 has its usual meaning, and qTC^0 denotes the class of functions computed by quasi-polynomial size TC^0 circuits.

The model of a Threshold Turing Machines (TTM) was introduced by Parberry and Schnitger [13]. A TTM is similar to an alternating machine, but instead of existential and universal states it has deterministic and threshold states, and it has a distinguished read-only guess tape. The successor configurations of a configuration in a threshold state all have the same state, but the initial segment of the guess tape through the position of the head is filled with zeroes and ones in all possible ways. Hence if the head on the guess tape is over the mth cell, there are 2^m successor configurations. The configuration is accepting if the majority of its successors are. A TTM also has a read-only input tape with random access via an index tape to allow for sub-linear runtimes. In the following, all TTMs are required to perform only constantly many threshold operations on each computation path. The following was noted by Allender [2]:

Proposition 1. *The class of languages accepted in time $O(t(n))$ on a TTM coincides with $TC^0(2^{O(t(n))})$, for every complexity function $t(n) = \Omega(\log n)$.*

Thus $TC^0(\tau_k(n))$ is equal to $\tau_{k-1}(\log n)$ time on a TTM, and in particular, TC^0 is equal to $O(\log n)$ time on a TTM, and polylogarithmic time on a TTM is the same as qTC^0.

The scheme of *concatenation recursion on notation* (CRN) was introduced by Clote [7]. We say that a function f is defined by CRN from g and h_0, h_1 if

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, s_0(y)) &= 2 \cdot f(\vec{x}, y) + h_0(\vec{x}, y) \qquad \text{for } y > 0 \\ f(\vec{x}, s_1(y)) &= 2 \cdot f(\vec{x}, y) + h_1(\vec{x}, y) \end{aligned}$$

provided that $h_i(\vec{x}, y) \le 1$ for all $\vec{x}, y$ and $i = 0, 1$.

Let $s_0(x) := 2x$, $s_1(x) := 2x + 1$, $|x| := \lceil \log_2(x + 1)\rceil$, $Bit(x, i) := \lfloor x/2^i \rfloor \bmod 2$, and for $j \le n$ let $\pi^n_j(x_1, \dots, x_n) := x_j$. Furthermore let $x\#_2 y := 2^{|x|\cdot|y|}$, and for $k \ge 2$ let $x\#_{k+1}y := 2^{|x|\#_k|y|}$. For $k \ge 2$, let $\mathcal{T}_k$ denote the least class of functions that contains the set

$$\{0, s_0, s_1, |.|, Bit, \cdot, \#_2, \dots, \#_k\} \cup \{\pi^n_j \,;\, j \le n\}$$

and is closed under composition and CRN. Clote and Takeuti [9] showed that $\mathcal{T}_2 = TC^0$. We generalize this to:

Theorem 2. *$\mathcal{T}_k = TC^0(\tau_k(n))$ for every $k \ge 2$. In particular, $\mathcal{T}_2 = TC^0$ and $\mathcal{T}_3 = qTC^0$.*

Proof. For the inclusion $\mathcal{T}_k \subseteq TC^0(\tau_k(n))$ the proof in [9] for the case $k = 2$ can be used to show that $TC^0(\tau_k(n))$ is closed under CRN. Then it is easy to see that the function $\#_k$ can be computed by circuits of the required size.

For the reverse inclusion, it is shown how to code the computation of a TTM operating in time $\tau_{k-1}(\log n)$ by a function in $\mathcal{T}_k$. This is done analogous to Clote's proof in [8] that the algebra A_0, which is $\mathcal{T}_2$ without multiplication, is equal to the alternating logarithmic time hierarchy. The idea is to code sequences of instructions instead of configurations and to use the closure of $\mathcal{T}_k$ under sharply bounded majority quantifiers. The details will be presented in the full version of the paper. □

3 First-Order Theories

For $k \ge 1$, the language L_k comprises the usual signature of arithmetic $0, S, +, \dot{-}, \cdot, \le$ plus function symbols for $\lfloor \frac{1}{2}x \rfloor$, $|x|$, $MSP(x, i) := \lfloor x/2^i \rfloor$ and, if $k \ge 2$, the functions $\#_2, \dots, \#_k$.

A quantifier of the form $\forall x \le t$, $\exists x \le t$ with x not occurring in t is called a *bounded quantifier*. Furthermore, the quantifier is called *sharply bounded* if the bounding term t is of the form $|s|$ for some term s. A formula is called (sharply) bounded if all quantifiers in it are (sharply) bounded.

We denote the class of quantifier-free formulas in L_k by $open_k$. The class of sharply bounded formulas in L_k is denoted $\Sigma^b_{0,k}$ or $\Pi^b_{0,k}$. For $i \in \mathbb{N}$, $\Sigma^b_{i+1,k}$ (resp. $\Pi^b_{i+1,k}$) is the least class containing $\Pi^b_{i,k}$ (resp. $\Sigma^b_{i,k}$) and closed under conjunction, disjunction, sharply bounded quantification and bounded existential (resp. universal) quantification. We usually omit the index k from the names of these classes, the value of k will always be clear from the context.

$BASIC_k$ denotes a set of quantifier-free axioms specifying the interpretations of the function symbols of L_k. It can most conveniently be taken as the set $BASIC$ from [3] together with the axioms for MSP and $\dot{-}$ from [16] and the two axioms

$$|x\#_j y| = S(|x|\#_{j-1}|y|)$$
$$z < x\#_j y \to |z| < |x\#_j y|$$

for $3 \le j \le k$.

For a class of formulas Φ, the axiom schema Φ-$LIND$ is

$$A(0) \wedge \forall x\,(A(x) \to A(Sx)) \to \forall x\, A(|x|)$$

for each $A(x) \in \Phi$. By Φ-IND we denote the usual induction scheme for formulas in Φ.

The theory S^i_k is the theory in the language L_k axiomatized by the $BASIC_k$ axioms and the Σ^b_i-$LIND$ scheme, and T^i_k is the theory given by $BASIC_k$ and Σ^b_i-IND. The results from [3, 4, 12, 11, 5] show a close connection between the theories S^i_2 and T^i_2 and polynomial time computations.

Before we can introduce the theories we are going to consider, we have to define some frequently used terms. Let

$$\begin{aligned}
2^{|x|} &:= 1\#_2\, x \\
mod2(x) &:= x \dot{-} 2 \cdot \lfloor \tfrac{1}{2} x \rfloor \\
Bit(x,i) &:= mod2(MSP(x,i)) \\
2^{\min(x,|y|)} &:= MSP(2^{|y|}, |y| \dot{-} x) \\
LSP(x,i) &:= x \dot{-} 2^{\min(i,|x|)} \cdot MSP(x,i) \\
\beta_a(w,i) &:= MSP(LSP(w, Si \cdot |a|), i \cdot |a|)
\end{aligned}$$

so that $LSP(x,|y|)$ returns the number consisting of the last $|y|$ bits of x, and if w codes a sequence $\langle b_1, \dots, b_\ell \rangle$ with $|b_i| \le |a|$ for all i, then $\beta_a(w,i) = b_i$. Thus the code for such a sequence is simply the number w whose binary representation consists of a 1 followed by the binary representations of the b_i concatenated, each padded with zeroes to be of exact length $|a|$. The replacement scheme $BB\Phi$ is then

$$\begin{aligned}
&\forall x \le |s|\, \exists y \le t(x)\, A(x,y) \to \\
&\qquad \exists w < 2(t^*\#_2\, 2s)\, \forall x \le |s|\, \beta_{t^*}(w,x) \le t(x) \\
&\qquad\qquad \wedge A(x, \beta_{t^*}(w,x))
\end{aligned}$$

for each $A(x,y) \in \Phi$, where $t^* := t^M(|s|)$ for some monotone term t^M that, provably in $BASIC_k$ + *open*-$LIND$, surpasses t. The comprehension scheme Φ-$COMP$ is

$$\exists y < 2^{|a|}\, \forall x < |a|\, (Bit(y,x) = 1 \leftrightarrow A(x))$$

for each $A(x) \in \Phi$.

The theory C^0_k is the theory in the language L_k given by $BASIC_k$, the *open*-$LIND$ scheme and $BB\Sigma^b_0$. The following proposition is easily proved:

Proposition 3. *C^0_k proves the Σ^b_0-$COMP$ axioms and the Σ^b_0-$LIND$ axioms.*

For a class of formulas Φ, a function f is said to be Φ-definable in a theory T if there is a formula $A(\vec{x}, y) \in \Phi$, describing the graph of f in the standard model, and a term $t(\vec{x})$, such that T proves

$$\forall \vec{x}\, \exists! y \le t(\vec{x})\, A(\vec{x}, y)\ .$$

In [10], the theory $\bar{R}^0_2$ was defined as S^0_2 plus Σ^b_0-$COMP$ and $BB\Sigma^b_0$, and it was shown that the Σ^b_1-definable functions of $\bar{R}^0_2$ are precisely the functions in TC^0. By Prop. 3, the theory C^0_2 is equal to $\bar{R}^0_2$, and thus the Σ^b_1-definable functions of C^0_2 are also the functions in TC^0. This can be generalized as follows:

Theorem 4. *The Σ^b_1-definable functions of C^0_k are exactly the functions in $TC^0(\tau_k(n))$.*

Proof. Using Proposition 3, we can do the same proof as in [10], showing that the Σ^b_1-definable functions of C^0_k are the function algebra $\mathcal{T}_k$, hence the result follows from Theorem 2 above. □

In particular, the Σ^b_1-definable functions of C^0_3 are the functions in qTC^0.

4 Counting Hierarchies

The counting hierarchy is the following hierarchy of functions: At the first level one has $1\#P := \#P$, the class of those functions computable as the number of accepting paths of an NP machine. The higher levels are defined inductively by $(i+1)\#P = \#P^{i\#P}$. The counting hierarchy is $FCH = \bigcup_{i\geq 1} i\#P$. We define $FCH(f(n))$ similarly to FCH except rather than using NP machines we use $O(f(n))$ time bounded nondeterministic machines. Another characterization of $FCH(f(n))$ is those functions computed by a TTM with runtime bounded by $O(f(n))$. If Ψ is a set of functions then $FCH(\Psi) = \cup_{f\in\Psi} FCH(f(n))$.

Definition: Let Ψ be a set of unary functions. A Ψ-sum is a sum of the form

$$\sum_{z=0}^{f(|x|)} g(x,z)$$

where f is in Ψ. We write Exp for the set $\{2^{n^k} ; k \in \mathbb{N}\}$.

Let $FCA(\Psi)$ denote the smallest class of functions that contains the arithmetic operations 0, 1, $+$, $\dot{-}$ and $\cdot$ and the projection functions π^n_j, and is closed under composition and Ψ-sums.

Corollary 4.4 in Vollmer and Wagner [18] shows that $FCA(Exp) = FCH$. Their proof generalizes in a straightforward manner to show:

Theorem 5. *Let Ψ be a set of complexity functions of at least polynomial growth. Then*

$$FCH(\Psi) = FCA(2^\Psi)$$

where $2^\Psi := \{2^f ; f \in \Psi\}$.

We define the class $FCA^{\vec{\alpha}}(\Psi)$ in the same way as $FCA(\Psi)$ except now we also let the predicate variables $\vec{\alpha}$ viewed as 0-1-valued functions be initial functions in the algebra. We define $CA^{\vec{\alpha}}(\Psi)$ to be the 0-1 valued functions in $FCA^{\vec{\alpha}}(\Psi)$.

Lemma 6. *Suppose $A(z,\vec{x},\vec{\alpha})$ is a predicate in $CA^{\vec{\alpha}}(2^{\tau_k(n)})$. Then*

$$f(y,\vec{x},\vec{\alpha}) = \mu z \leq y\, A(z,\vec{x},\vec{\alpha})$$

is in $FCA^{\vec{\alpha}}(2^{\tau_k(n)})$.

5 Second-Order Theories

Let $\mathcal{L}_k$ be the language L_k extended to allow second-order unary predicate variables α^t_i for $i \in \mathbb{N}$ and L_k-term t. The idea is t is a bound on the range of true values of this variable. A second-order formula is called *bounded* if all its first-order quantifiers are bounded. We usually omit the index i and use other lower case Greek letters as names for predicate variables instead.

$\Sigma^{1,b}_{0,k} = \Pi^{1,b}_{0,k}$ is the class of formulas with only bounded first-order quantifiers. Then for every i, the class $\Sigma^{1,b}_{i+1,k}$ ($\Pi^{1,b}_{i+1,k}$) is the least class that contains $\Pi^{1,b}_{i,k}$ (resp. $\Sigma^{1,b}_{i,k}$) and is closed under conjunction, disjunction, bounded first-order quantification and existential (resp. universal) second-order quantification. A formula B is a $\Delta^{1,b}_{i,k}$ in a theory T if B is provably in T equivalent to both a $\Sigma^{1,b}_{i,k}$ and a $\Pi^{1,b}_{i,k}$-formula.

We will use the following boundedness axioms for predicate variables in our theories:

$$\forall x\,(\alpha^t(x) \to x < t)\ .$$

We write $\{x\}V^t$ for the abstract $\{x\}(x \leq t \wedge V)$. Let $\langle\, ,\rangle$ be a pairing function with inverses $()_1$ and $()_2$, and let $bd(s,t)$ be a term that bounds all pairs of the form $\langle i,j\rangle$ where $i \leq t$ and $j \leq s$. Let $\boldsymbol{\beta}(b,\alpha)$ be the abstract for the second-order β function $\{x\}\alpha(\langle b,x\rangle)$, and let $S\alpha$ be the abstract $\{x\}\big[\alpha(x) \leftrightarrow \exists y \leq x \neg\alpha(y)\big]$. Then the counting axiom is given by

$$\begin{aligned}\exists\varphi^{bd(t,|t|)}\, \forall j \leq |t|\, (\boldsymbol{\beta}(0,\varphi)(j) \leftrightarrow j = 0 \wedge \alpha^t(0))\\ \wedge\ \forall i < t\, \big[\big(\neg\alpha^t(Si) \wedge \boldsymbol{\beta}(Si,\varphi) \equiv_{|t|} \boldsymbol{\beta}(i,\varphi)\big)\\ \vee\ \big(\alpha^t(Si) \wedge \boldsymbol{\beta}(Si,\varphi) \equiv_{|t|} S\boldsymbol{\beta}(i,\varphi)\big)\big]\end{aligned}$$

where $\alpha_1 \equiv_t \alpha_2 := \forall j \leq t\, \alpha_1(j) \leftrightarrow \alpha_2(j)$.

Definition:

1. Φ-BCA, Φ-bounded comprehension axiom is the following scheme:

$$\exists\alpha^t\, \forall x \leq t\, (\alpha(x) \leftrightarrow A(x))$$

   where A is in Φ and does not contain the variable α.

2. Φ-BCR, Φ-bounded comprehension rule is the following inference:

$$\frac{\Gamma \Rightarrow A(V^t), \Delta}{\Gamma \Rightarrow \exists\varphi^t\, A(\varphi^t), \Delta}$$

   where V is a Φ-abstract.

3. Φ-AC, Φ-second-order replacement is the following scheme

$$\begin{aligned}\forall x \leq s\, \exists\alpha^t\, B(x,\alpha)\\ \leftrightarrow\ \exists\psi^u\, \forall x \leq s\, B(x,\boldsymbol{\beta}(x,\psi^u))\end{aligned}$$

   where A is in Φ and $u := bd(s,t)$.

4. Φ-ACR, Φ-second-order replacement rule is the following inference

$$\frac{\Gamma \Rightarrow \forall x \leq s\, \exists\varphi^t\, A(x,\varphi^t), \Delta}{\Gamma \Rightarrow \exists\psi^u\, \forall x \leq s\, A(x,\boldsymbol{\beta}(x,\psi^u)), \Delta}$$

   where A is in Φ and $u := bd(s,t)$.

The theories $\mathbf{D}_k^0$ over the language $\mathcal{L}_k$ are axiomatized as $BASIC_k$ together with counting axioms, *open-IND*, *open-BCA* and $\Sigma_0^{1,b}$-*AC*. We could have alternatively characterized this theory as those statements provable in the second-order sequent calculus with $BASIC_k$ axioms, counting axioms, *open-IND*, $\Sigma_0^{1,b}$-*ACR*, and *open-BCR*.

Theorem 7. *$\mathbf{D}_k^0$ proves the following extensionality axioms, where $u \geq \max(s,t)$:*

$$\forall x \leq u\, (\alpha^s(x) \leftrightarrow \gamma^t(x)) \rightarrow \forall x\, (\alpha^s(x) \leftrightarrow \gamma^t(x))\,.$$

This follows immediately from the boundedness axiom. Using an abstract to code a pair of predicates into a single predicate, one can show the following:

Theorem 8. *$\mathbf{D}_k^0$ proves $\Sigma_1^{1,b}$-AC.*

Similar to Proposition 3, we have

Lemma 9. *$\mathbf{D}_k^0$ proves $\Delta_1^{1,b}$-BCA and $\Delta_1^{1,b}$-IND.*

Proof. Let $A(x)$ be $\Delta_1^{1,b}$ in $\mathbf{D}_k^0$, and consider the formula $A(x) \leftrightarrow \alpha^0(0)$, which is equivalent in $\mathbf{D}_k^0$ to a $\Sigma_1^{1,b}$-formula $B(x,\alpha^0)$. Now $\mathbf{D}_k^0$ proves $\forall x \leq t\, \exists \alpha^0\, B(x,\alpha^0)$, hence by $\Sigma_1^{1,b}$-*AC* there is a predicate $\psi^{bd(t,0)}$ such that $\forall x \leq t \beta(x,\psi)(0) \leftrightarrow A(x)$, and hence by *open-BCA* there is φ^t such that $\forall x \leq t\, \varphi^t(x) \leftrightarrow A(x)$, which proves $\Delta_1^{1,b}$-*BCA*. Now $\Delta_1^{1,b}$-*IND* follows immediately from $\Delta_1^{1,b}$-*BCA* and *open-IND*. □

Using $\Delta_1^{1,b}$-*BCA*, it is possible to show:

Theorem 10. *$\mathbf{D}_k^0$ can $\Sigma_1^{1,b}$-define the functions in $FCA^{\vec{\alpha}}(2^{\tau_k(n)})$. Moreover, $\mathbf{D}_k^0$ can $\Sigma_1^{1,b}$-define any $f \in FCA(2^{\tau_k(n)})$ using a formula not containing free predicate variables.*

Proof. The only nontrivial thing to prove is the closure of the $\Sigma_1^{1,b}$-definable functions under summation. So let $g(x)$ be $\Sigma_1^{1,b}$-definable in $\mathbf{D}_k^0$, and let s be a term bounding g. Now $y < g(x)$ is $\Delta_1^{1,b}$ in $\mathbf{D}_k^0$, so by $\Delta_1^{1,b}$-*BCA* we can define a predicate $\alpha^{bd(s,t)}$ with

$$\forall x,y \leq t\, \alpha(\langle y,x\rangle) \leftrightarrow y < g(x)\ .$$

Now note that the number of $x \leq bd(s,t)$ with $\alpha(x)$ is $\sum_{i=0}^t g(i)$, and this number can be counted by use of the counting axiom. □

This implies also that every predicate in $CA^{\vec{\alpha}}(2^{\tau_k(n)})$ is $\Delta_1^{1,b}$ in $\mathbf{D}_k^0$.

6 A Witnessing Argument

The following closure properties of $\Delta_1^{1,b}$ formulas in $\mathbf{D}_k^0$ are easily verified.

Lemma 11. *The class of $\Delta_1^{1,b}$-formulas in $\mathbf{D}_k^0$ is closed under boolean combinations, bounded first-order quantification, substitution of $\Delta_1^{1,b}$-abstracts for free predicate variables and substitution of terms containing $\Sigma_1^{1,b}$-defined functions for free first-order variables.*

Let $\tilde{\Sigma}_1^{1,b}$ be the class consisting of formulas of the form $\exists x \leq t\, \exists \varphi\, A$ or of the form $\forall x \leq t\, \exists \varphi\, A$ where A is $\Sigma_0^{1,b}$. Suppose $\mathbf{D}_k^0$ defines some function f by proving $\forall x\, \exists y\, \exists \varphi\, A$ where A is in $\Sigma_0^{1,b}$. Then by Parikh's Theorem, $\mathbf{D}_k^0$ proves $\Rightarrow \exists y \leq t\, \exists \varphi\, A$ and given the form of $\mathbf{D}_k^0$'s axioms and rules of inference, by cut-elimination we can assume all sequents in this proof contain only $\tilde{\Sigma}_1^{1,b}$-formulas. We define a witnessing predicate for $\tilde{\Sigma}_1^{1,b}$-formulas as follows:

1. If $A(\vec{a},\vec{\alpha}) \in \Sigma_0^{1,b}$ then $Wit2_A(\gamma,\vec{a},\vec{\alpha}) := A(\vec{a},\vec{\alpha})$.
2. If $A(\vec{a},\vec{\alpha})$ is of the form $\exists \varphi^t\, B$ where $B \in \Sigma_0^{1,b}$, then
$$Wit2_A(\gamma^t,\vec{a},\vec{\alpha})\ :=\ B(\gamma^t,\vec{a},\vec{\alpha}).$$
3. If $A(\vec{a},\vec{\alpha})$ is of the form $\exists x \leq s\, \exists \varphi^t\, B$ where $B \in \Sigma_0^{1,b}$, then
$$Wit2_A(\gamma^t,\vec{a},\vec{\alpha})\ :=\ \exists x \leq s\, B(\gamma^t,x,\vec{a},\vec{\alpha})$$
4. If $A(\vec{a},\vec{\alpha})$ is of the form $\forall x \leq s\, \exists \varphi^t\, B$ where $B \in \Sigma_0^{1,b}$, then $Wit2_A(\gamma^{bd(s,t)},\vec{a},\vec{\alpha})$ is
$$\forall x \leq s\, B(\beta(x+1,\gamma^{bd(s,t)}),x,\vec{a},\vec{\alpha})\ .$$

Lemma 12. *Let $A(\vec{a},\vec{\alpha})$ be a $\tilde{\Sigma}_1^{1,b}$-formula. Then $\mathbf{D}_k^0$ proves*

$$A(\vec{a},\vec{\alpha}) \leftrightarrow \exists \psi^t\, Wit2_A(\psi^t,\vec{a},\vec{\alpha})$$

The statement is trivial if A falls under the first three cases listed above. For the fourth case it follows by $\Sigma_1^{1,b}$-*AC*.

Lemma 13. *Any $\Sigma_0^{1,b}$-formula with free variables among $\gamma^t,\vec{\alpha}$ is in $CA^{\gamma^t,\vec{\alpha}}(2^{\tau_k(n)})$. In particular, for a $\tilde{\Sigma}_1^{1,b}$-formula $A(\vec{a},\vec{\alpha})$, $Wit2_A(\gamma^t,\vec{a},\vec{\alpha})$ is a predicate in $CA^{\gamma^t,\vec{\alpha}}(2^{\tau_k(n)})$.*

For a cedent of $\tilde{\Sigma}_1^{1,b}$-formulas $\Gamma = A_1,\ldots,A_n$ we define $Wit2_{\wedge\Gamma}(\gamma^{t_\Gamma},\vec{a},\vec{\alpha})$ to be

$$\bigwedge_i Wit2_{A_i}(\beta(i,\gamma^{t_\Gamma}),\vec{a},\vec{\alpha})\ ,$$

where $t_\Gamma := bd(n, max(t_1, ..., t_n))$ and t_i is the bound on the witnessing predicate for A_i. Likewise, we define $Wit2_{\vee\Gamma}(\gamma^{t_\Gamma}, \vec{a}, \vec{\alpha})$ to be

$$\bigvee_i Wit2_{A_i}(\beta(i, \gamma^{t_\Gamma}), \vec{a}, \vec{\alpha}) .$$

Theorem 14. *Suppose $\mathbf{D}^0_k \vdash \Gamma \Rightarrow \Delta$ where Γ and Δ are cedents of $\tilde{\Sigma}^{1,b}_1$-formulas having free variables among $\vec{c}, \vec{\gamma}$. Then there is a predicate $M^{t'}$ in $CA^{\alpha^{t_\Gamma}, \vec{\gamma}}(2^{\tau_k(n)})$ which is $\Delta^{1,b}_1$ in $\mathbf{D}^0_k$ such that:*

$$\mathbf{D}^0_k \vdash Wit2_{\wedge\Gamma}(\alpha^{t_\Gamma}, \vec{c}, \vec{\gamma}) \rightarrow Wit2_{\vee\Delta}(\{x\}M^{t_\Delta}(x, \vec{c}, \alpha^{t_\Gamma}, \vec{\gamma}), \vec{c}, \vec{\gamma}).$$

Proof. We can assume that any $\mathbf{D}^0_k$ proof of a sequent of $\tilde{\Sigma}^{1,b}_1$-formulas contains only $\tilde{\Sigma}^{1,b}_1$-formulas. Also, we can assume that no predicate variable on the right hand side of a sequent in the proof is eliminated by an second-order existential introduction or *open-BCR* inference, because otherwise we could replace it everywhere by the abstract $\{x\}(1 = 1)$ and add some weakenings to make the resulting figure a valid proof.

The proof proceeds by induction on a $\mathbf{D}^0_k$ sequent calculus proof of $\Gamma \Rightarrow \Delta$. The induction base is trivial for the logical, $BASIC$ and boundedness axioms since these consist of $\Sigma^{1,b}_0$-formulas. For the counting axiom, note that the predicate $C^{bd(t,|t|)}(x, \alpha^t)$ defined by

$$Bit\Big(\sum_{j=0}^{(x)_1} \alpha^t(j)\, ,\, (x)_2\Big) = 1$$

is in $CA^{\alpha^t}(2^{\tau_k(n)})$ and hence $\Delta^{1,b}_1$ in $\mathbf{D}^0_k$, and that $C^{bd(t,|t|)}(x, \alpha^t)$ witnesses the counting axiom. For the induction step, we only treat the cases where the last inference is a right bounded-universal introduction, *open-BCR*, *open-IND* or $\Sigma^{1,b}_0$-ACR. For sake of readability we also do not display the free variables $\vec{c}, \vec{\gamma}$.

Suppose the last inference is

$$\frac{b \leq t, \Gamma \Rightarrow A(b), \Delta}{\Gamma \Rightarrow \forall x \leq t\, A(x), \Delta} .$$

By the hypothesis there is a $\Delta^{1,b}_1$-abstract $M_1^{t_{A,\Delta}}$ in $CA^{\alpha^r}(2^{\tau_k(n)})$ such that

$$\mathbf{D}^0_k \vdash Wit2_{b \leq t \wedge \Gamma}(\alpha^r, b) \rightarrow Wit2_{A \vee \Delta}(\{x\}M_1^{t_{A,\Delta}}(x, b, \alpha^r), b) ,$$

where r is $t_{b\leq t,\Gamma}$. Now either A is in $\Sigma^{1,b}_0$, or A is of the form $\exists\varphi\, B$ where B is $\Sigma^{1,b}_0$. In both cases let $\tilde{\alpha}^r := \{x\}\, \alpha^r\big(\langle (x)_1 \dot{-} 1, (x)_2\rangle\big)$, and let $g(x) = \mu y \leq t\, \neg Wit2_A(\beta(1, \{x\}M_1(x, y, \tilde{\alpha}^r)), y)$, which is in $FCA^{\alpha^r}(2^{\tau_k(n)})$ by Lemma 6. In the first case, we define

$$M^s(x, \alpha^r) := M_1(x, g(x), \tilde{\alpha}^r)$$

where $s = t_{(\forall x \leq t) A(x, \vec{c}), \Delta}$. If $g(x) < t + 1$ this abstract will provide a witness to Δ. Otherwise, notice that $\forall x \leq t\, A$ is a true $\Sigma^{1,b}_0$-formula so any $\Delta^{1,b}_1$-abstract in $CA^{\alpha^r, \vec{\gamma}}(2^{\tau_k(n)})$ witnesses the succedent. So

$$\mathbf{D}^0_k \vdash Wit2_{\wedge\Gamma}(\alpha^r) \rightarrow Wit2_{\forall x \leq tA \vee \Delta}(\{x\}M^s(x, \alpha^r)) .$$

In the second case, A is of the form $(\exists\varphi)B$ where $B \in \Sigma^{1,b}_0$. Let m be the number of formulas in the lower succedent. We define $M^s(x, \alpha^r)$ where s is as before to be

$$((x)_1 = 1 \wedge M_1(\langle 1, ((x)_2)_2\rangle, ((x)_2)_1, \tilde{\alpha}^r,)) \vee (2 \leq (x)_1 \leq m \wedge M_1(x, g(x), \tilde{\alpha}^r)))$$

Now either $\forall x \leq t\, A(x)$ holds or there is some $b \leq t$ that $\neg A(b)$. In the first case, the $\beta(1, M^s)$ witnesses $\forall x \leq t\, A(x)$. Otherwise Δ is witnessed by the rest of M^s. So

$$\mathbf{D}^0_k \vdash Wit2_{\wedge\Gamma}(\alpha^r) \rightarrow Wit2_{\forall x \leq tA \vee \Delta}(\{x\}M^s(x, \alpha^r)).$$

Suppose the final inference is an *open-BCR*

$$\frac{\Gamma \Rightarrow A(V^t), \Delta}{\Gamma \Rightarrow (\exists\varphi^t)A(\varphi), \Delta}$$

where V^t is an open-abstract. By hypothesis there is a $\Delta^{1,b}_1$-abstract $M_1^{t_{A,\Delta}} \in CA^{\alpha^r}(2^{\tau}_k(n))$ such that

$$\mathbf{D}^0_k \vdash Wit2_{\wedge\Gamma}(\alpha^r) \rightarrow Wit2_{A \vee \Delta}(\{x\}M_1^{t_{A,\Delta}}(x, \alpha^r)),$$

where r is t_Γ. Let $M^s(x, \alpha^r)$ be

$$((x)_1 = 1 \wedge V^t((x)_2) \vee ((x)_1 > 1 \wedge M_1^{t_{A,\Delta}}(x, \alpha_r))$$

where $s := bd(m + 1, max(t, t_{A,\Delta})$. It is now easy to see that

$$\mathbf{D}^0_k \vdash Wit2_{\wedge\Gamma}(\alpha^r) \rightarrow Wit2_{(\exists\varphi)A \vee \Delta}(\{x\}M^s(x, \alpha^r)).$$

Suppose the final inference is an *open-IND*:

$$\frac{A(y), \Gamma \Rightarrow A(Sy), \Delta}{A(0), \Gamma \Rightarrow A(t), \Delta}$$

By induction there is a $\Delta_1^{1,b}$-predicate $M_1^s \in CA^{\alpha^r}(2^{\tau_k(n)})$ such that

$$\mathbf{D}_k^0 \vdash Wit2_{A(y)\wedge\Gamma}(\alpha^r, y) \\ \rightarrow Wit2_{A(Sy)\vee\Delta}(\{x\}M_1^s(x,y,\alpha^r), y),$$

where $r = t_{A(y),\Gamma}$ and $s = t_{A(Sy),\Delta}$. Since $A(y)$ is open, we can define a function $g(x,\alpha^r) = \mu y < t\, \neg A(Sy)$, so that $g \in FCA^{\alpha^r}(2^{\tau_k(n)})$ by Lemma 6. Now define M to be

$$M^s := M_1^s(x, g(x,\alpha^r), \alpha^r).$$

By $\Sigma_0^{1,b}$-IND either $A(t)$ holds or $\neg A(0)$ holds or g returns a value such that $A(y)$ and $\neg A(Sy)$. In the first two cases, M^s trivially witnesses the succedent. In the last case, by the induction hypothesis M^s will produce a witness for some formula in Δ.

For the case where the final inference is $\Sigma_0^{1,b}$-ACR, note that $Wit2_{\exists\psi\,\forall x\lessdot t\, A(x,\boldsymbol{\beta}(x,\psi))}$ and $Wit2_{\forall x\lessdot t\,\exists\varphi\, A(x,\varphi)}$ are the same predicate, so any abstract witnessing the upper sequent will also witness the lower sequent. □

¿From the witnessing theorem we get the following result immediately.

Theorem 15. *Suppose $A(\vec{x},y)$ is a $\Sigma_1^{1,b}$-formula where $\vec{x},y$ are all the free variables of A such that $\mathbf{D}_k^0 \vdash \forall\vec{x}\,\exists y\, A(\vec{x},y)$. Then there is a $\Sigma_1^{1,b}$-formula $B(\vec{x},y)$, a term t and a function $f \in FCA(2^{\tau_k(n)})$ so that*

1. $\mathbf{D}_k^0 \vdash \forall\vec{x}\,\forall y\, B(\vec{x},y) \rightarrow A(\vec{x},y)$
2. $\mathbf{D}_k^0 \vdash \forall\vec{x}\,\exists! y \leq t\, B(\vec{x},y)$
3. *For all $\vec{n}$, $\mathbb{N} \models B(\vec{n}, f(\vec{n}))$*

In particular, this implies that any $\Sigma_{1,k}^{1,b}$-definable function of $\mathbf{D}_k^0$ is in $FCA(2^{\tau_k(n)})$. Together with Theorem 10, this gives the characterization of the $\Sigma_{1,k}^{1,b}$-definable functions in $\mathbf{D}_k^0$.

7 $RSUV$-isomorphism

First we define a translation mapping every L_{k+1}-formula A to a $\mathcal{L}_k$-formula A^H. The translation is essentially the same as the one defined in [15, 16].

Inductively, we define for each L_{k+1}-term t a $\Delta_1^{1,b}$-formula $A_t(x)$ and a L_k-term T_t. Then t^H is the abstract $\{x\}(x \leq T_t \wedge A_t(x))$. The idea is that the value of t is $\sum_{i=0}^{T_t} A_t(i)2^i$, i.e. A_t codes t in binary.

First, $T_0 := 0$ and A_0 is $0 = 1$. For a variable a, $T_a := a$ and A_a is a second-order variable α^a. Then for each function symbol f, $T_{f(\bar{t})}$ and $A_{f(\bar{t})}$ are defined according to the computation of the bits of $f(\bar{t})$ from the bits of $\bar{t}$. E.g. $T_{\lfloor\frac{1}{2}t\rfloor} := T_t \dot{-} 1$ and $A_{\lfloor\frac{1}{2}t\rfloor}(x) := A_t(x+1)$, and $T_{St} := T_t + 1$ and

$$A_{St}(x) :=\ A_t(x) \leftrightarrow \exists y \leq x\, \neg A_t(y)\ .$$

The most intriguing case is multiplication. First let $2^y\alpha(x)$ be $x \geq y \wedge \alpha(x \dot{-} y)$, and let $(\alpha +^H \beta)$ be an abstract such that $A_{s+t}(x)$ is $(s^H +^H t^H)(x)$. Now we define $T_{s\cdot t} := T_s + T_t$, and $A_{s\cdot t}$ as

$$\exists\gamma^m\, Table(s^H, t^H, \gamma_m, T_s) \wedge \boldsymbol{\beta}(T_s+1, \gamma^m)(x)$$

where $m := bd(T_s+1, T_{s\cdot t})$ and the formula $Table(s^H, t^H, \gamma, a)$ is defined as

$$\forall y \leq T_{s\cdot t}\, \neg\boldsymbol{\beta}(0,\gamma)(y) \wedge \\ \forall y \leq a\, \Big(\big(\neg s^H(y) \wedge \boldsymbol{\beta}(Sy,\gamma) \equiv_{T_{s\cdot t}} \boldsymbol{\beta}(y,\gamma)\big) \vee \\ \big(s^H(y) \wedge \boldsymbol{\beta}(Sy,\gamma) \equiv_{T_{s\cdot t}} (\boldsymbol{\beta}(y,\gamma) +^H 2^y t^H)\big)\Big)$$

i.e. γ codes the computation of $s \cdot t$ as the sum of the vector of numbers $t \cdot Bit(s,i) \cdot 2^i$ for $i < |s|$.

To see that $A_{s\cdot t}$ is $\Delta_1^{1,b}$ we need to prove in $\mathbf{D}_k^0$ that given s^H, t^H and $a \leq T_s{+}1$, there is a unique predicate $\gamma^{bd(a+1,T_{s\cdot t})}$ such that $Table(s^H, t^H, \gamma, a)$ holds. Then the $\Sigma_1^{1,b}$-formula $A_{s\cdot t}$ is equivalent to the $\Pi_1^{1,b}$-formula

$$\forall\gamma^m\, Table(s^H, t^H, \gamma^m, T_s) \rightarrow \boldsymbol{\beta}(T_s+1, \gamma^m)(x).$$

Now the existence of γ is proved from the counting axiom by formalizing in $\mathbf{D}_k^0$ a reduction of vector summation to counting such as the one in [6], and the uniqueness follows from extensionality.

The definitions for the other function symbols can be found in [15, 16]. To define A^H for a formula A, first $(s \leq t)^H$ is defined as $s^H \leq_H t^H$, where $\leq H$ expresses the lexicographic ordering of predicates. Then $(s = t)^H$ is $(s \leq t)^H \wedge (t \leq s)^H$. The translation commutes with the propositional connectives. For quantified formulas note that $B(a)^H$ is of the form $B^H(a, \alpha^a)$. Now $(\exists x \leq t\, B(x))^H$ is

$$\exists x \leq T_t\, \exists\varphi^x\, \left(x^H \leq_H t^H \wedge B^H(x,\varphi^x)\right)$$

and $(\forall x \leq t\, B(x))^H$ is

$$\forall x \leq T_t\, \forall\varphi^x\, \left(x^H \leq_H t^H \rightarrow B^H(x,\varphi^x)\right)\ ,$$

where A_x is φ^x. Finally define $(\exists x\ B(x))^H$ as $\exists x\exists\varphi^x B^H(x,\varphi^x)$ and $(\forall x B(x))^H$ as $\forall x\forall\varphi^x B^H(x,\varphi^x)$.

Theorem 16. *If $C_{k+1}^0 \vdash A$, then $\mathbf{D}_k^0 \vdash A^H$, for every L_{k+1}-formula A.*

Proof. By induction on the length of a proof of A in C^0_{k+1}. The translations of $BASIC$ axioms can all be proved in $\mathbf{D}^0_k$ by use of $\Delta^{1,b}_1$-IND, where for those axioms concerning multiplication the counting axiom has to be applied. The translation of *open*-$LIND$ are proved by $\Delta^{1,b}_1$-IND, and the translation of $BB\Sigma^b_0$ is proved by use of $\Sigma^{1,b}_1$-AC. □

Next we define a translation mapping a $\mathcal{L}_k$-formula B to a L_{k+1}-formula B^L. The translation is the same used in [16].

For a term t, t^L is constructed by replacing every variable a in t by $|a|$. Then $(s = t)^L$ is $s^L = t^L$ and $(s \leq t)^L$ is $s^L \leq t^L$. For $A = \alpha^t(s)$, A^L is defined as $s^L \leq t^L \wedge Bit(a, s^L) = 1$. The translation commutes with the propositional connectives. For the quantifiers, we have three cases:

- If A is $\forall x\, B$ or $\exists x\, B$, then A^L is simply $\forall x\, B^L$ resp. $\exists x\, B^L$.
- If A is $\forall \varphi^t\, B(\varphi^t)$ or $\exists \varphi^t\, B(\varphi^t)$, then A^L is $\forall x < 2^{t^L+1}\, B^L(x)$ resp. $\exists x < 2^{t^L+1}\, B^L(x)$.
- If A is $\forall x \leq t\, B$ or $\exists x \leq t\, B$, and B^L is $\tilde{B}(|x|)$, then A^L is $\forall x \leq t^L\, \tilde{B}(x)$ resp. $\exists x \leq t^L\, \tilde{B}(x)$.

Note that due to the presence of the function $\#_{k+1}$ every term of the form t^L for L_k-term t can be written in the form $|s|$ for some L_{k+1}-term s. Hence the bound 2^{t^L+1} can be expressed by a term, and the translations of first-order bounded quantifiers are sharply bounded, which gives the following crucial property of the translation.

Lemma 17. *If B is a $\Sigma^{1,b}_{i,k}$-formula, then A^L is equivalent to a $\Sigma^b_{i,k+1}$-formula in C^0_{k+1}.*

Theorem 18. *If $\mathbf{D}^0_k \vdash B$, then $C^0_{k+1} \vdash B^L$, for every $\mathcal{L}_k$-formula B.*

Proof. By induction on the length of a proof of B in $\mathbf{D}^0_k$. Note that $BASIC$ axioms are translated to instances of $BASIC$ axioms and the translation of the boundedness axiom is tautological. Applications of *open*-IND and *open*-BCA are provable by *open*-$LIND$ and *open*-$COMP$ respectively, where the latter is provable in C^0_{k+1} by Lemma 3. The translation of $\Sigma^{1,b}_0$-AC is provable by use of $BB\Sigma^b_0$. Finally the translation of the counting axiom can be proved in C^0_{k+1} by use of the reduction of counting to multiplication in [6]. □

Finally, we show that the translations ^{H} and ^{L} are inverse to each other. There are very easy translations * from L_{k+1} to itself and $^\square$ from $\mathcal{L}_k$ to itself such that the following holds.

Theorem 19. *1. $C^0_{k+1} \vdash A \leftrightarrow A^{HL*}$ for every L_{k+1}-formula A.*

2. $\mathbf{D}^0_k \vdash B \leftrightarrow B^{LH\square}$, for every bounded $\mathcal{L}_k$-formula B.

Proof. For both statements, one direction follows by applying Theorems 16 and 18 in succession. The other direction is by induction on the complexity of A or B. The proof is the same as in [16]. □

This together with Theorems 16 and 18 immediately yields the following.

Corollary 20. *1. For every L_{k+1}-formula A, $C^0_{k+1} \vdash A$ iff $\mathbf{D}^0_k \vdash A^H$.*

2. For every bounded $\mathcal{L}_k$-formula B, $\mathbf{D}^0_k \vdash B$ iff $C^0_{k+1} \vdash B^L$.

Acknowledgments

We would like to thank the following people: Jan Krajíček and an anonymous referee for the paper [10] suggested that $\bar{R}^2_0 = C^0_2$ might be $RSUV$-isomorphic to a certain subtheory of $\mathbf{D}^0_1$, which was the starting point of this paper. Peter Clote brought [18] to our attention, and Eric Allender referred us to the concept of Threshold Turing Machines and his [2].

References

[1] E. Allender. A note on the power of threshold circuits. In *Proceedings of the 30th FOCS*, pages 580–584, 1989.

[2] E. Allender. The permanent requires large uniform threshold circuits. Manuscript. Preliminary Version appeared in COCOON'96, 1996.

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

[4] S. R. Buss. Axiomatizations and conservation results for fragments of bounded arithmetic. In *Logic and Computation*, volume 106 of *Contemporary Mathematics*, pages 57–84. American Mathematical Society, Providence, 1990.

[5] S. R. Buss and J. Krajíček. An application of boolean complexity to separation problems in bounded arithmetic. *Proceedings of the London Mathematical Society — 3rd Series*, 69:1–21, 1994.

[6] A. C. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM Journal of Computing*, 13:423–439, 1984.

[7] P. Clote. Sequential, machine independent characterizations of the parallel complexity classes $ALogTIME$, AC^k, NC^k and NC. In S. R. Buss and P. J. Scott, editors, *Feasible Mathematics*, pages 49–69. Birkhäuser, Boston, 1990.

[8] P. Clote. Computation models and function algebras. to appear in E. Griffor (ed.) *Handbook of Recursion Theory*, 1996.

[9] 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.

[10] J. Johannsen. A bounded arithmetic theory for constant depth threshold circuits. In P. Hájek, editor, *GÖDEL '96*, pages 224–234, 1996. Springer Lecture Notes in Logic 6.

[11] J. Krajíček. Fragments of bounded arithmetic and bounded query classes. *Transactions of the AMS*, 338:587–598, 1993.

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

[13] I. Parberry and G. Schnitger. Parallel computation with threshold functions. *Journal of Computer and System Sciences*, 36:278–302, 1988.

[14] A. A. Razborov. An equivalence between second order bounded domain bounded arithmetic and first order bounded arithmetic. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, volume 23 of *Oxford Logic Guides*, pages 247–277. Clarendon Press, Oxford, 1993.

[15] G. Takeuti. S^i_3 and $\overset{\circ}{V}{}^i_2\,(BD)$. *Archive for Mathematical Logic*, 29:149–169, 1990.

[16] 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.

[17] S. Toda. On the computational power of PP and $\oplus P$. In *Proceedings of the 30th FOCS*, pages 26–35, 1989.

[18] H. Vollmer and K. Wagner. Recursion theoretic characterizations of complexity classes of counting functions. *Theoretical Computer Science*, 163:245–258, 1996.