On the Δ_1^b-Bit-Comprehension Rule

Jan Johannsen[1,⋆] and Chris Pollett[2]

[1] Dept. of Mathematics
U.C. San Diego
[2] Dept. of Mathematics and Computer Science
Clark University

1 Introduction

The Δ_1^b-bit-comprehension rule roughly states the following: Given n and a predicate $A(x)$ that has been proven to be Δ_1^b, i.e., equivalent to both an NP- (Σ_1^b-) and a *co*-NP- (Π_1^b-) predicate, there is a number w of length n such that for every $i < n$, the ith bit of w is set if and only if $A(i)$ holds. One can think of w as coding the set of small i such that $A(i)$ holds.

We consider the theory of Bounded Arithmetic Δ_1^b-CR that has this rule as its main axiom. This theory is related to the computational complexity class TC^0 of functions computable by constant-depth threshold circuits. We show that the theory C_2^0 of [9], whose Σ_1^b-definable functions are TC^0, is $\forall\Sigma_1^b$-conservative over Δ_1^b-CR.

Theories of Bounded Arithmetic that correspond to the complexity class TC^0 have been described earlier by the authors [9, 8] as well as by Clote and Takeuti [7]. So why do we come up with yet another one? We think there are two reasons that make Δ_1^b-CR more interesting than the previous theories for TC^0.

First, one can argue that it is the weakest natural theory whose Σ_1^b-definable functions are TC^0, as the closure of the Σ_1^b-definable functions under concatenation recursion on notation (CRN) is essentially the same as Δ_1^b-comprehension.

Second, we will show that Δ_1^b-CR has a tighter connection to TC^0 than the previously considered theories: The Σ_2^b-theorems of Δ_1^b-CR can be witnessed by counterexample computations (a concept introduced by [13, 11] that we will define below) where the Student has the computational capabilities of TC^0. Similar to the results of [12], this will allow us to show that a collapse of stronger theories, S_2^1 or R_2^1, to Δ_1^b-CR implies that every NP-predicate can be decided by non-uniform TC^0-circuits.

2 Uniform and Non-Uniform TC^0

A *threshold circuit* is a circuit built up from boolean variables and their negations by threshold gates of the form $T_k(x_1, \ldots, x_m)$, where the boolean function T_k is

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

defined by

$$T_k(x_1, \ldots, x_m) := \begin{cases} 1 \text{ if } \#\{\, i \,;\, x_i = 1 \,\} \geq k \\ 0 \text{ otherwise} \end{cases} .$$

If the variables in the circuit are $x_1, \ldots, x_n$, then it computes a boolean function $\{0,1\}^n \to \{0,1\}$. More generally, we can let it compute a function $\{0,1\}^n \to \{0,1\}^m$ by allowing several outputs.

A boolean function $f : \{0,1\}^* \to \{0,1\}^*$ is computed by a circuit family $\langle C_n ; n \in \mathbb{N} \rangle$ if for each n, C_n computes $f|_{\{0,1\}^n}$. The non-uniform class TC^0 is defined as the class of functions computable by a family of threshold circuits of polynomial size and constant depth, i.e., there are a polynomial $p(n)$ and a constant d such that for all n, $size(C_n) \leq p(n)$ and $depth(C_n) \leq d$.

Non-uniform circuit families can compute functions that are not computable. For example, let K be an undecidable set of natural numbers, then the characteristic function of $\{\, 1^k \,;\, k \in K \,\}$ is computable by a trivial circuit family of linear size and depth 1. To overcome this sometimes unwanted feature, circuit families are required to satisfy certain uniformity conditions. For TC^0-circuits, the most suitable uniformity notion is DLogTime-uniformity, see [3] for the somewhat involved definition.

DLogTime-uniform TC^0 is a fairly natural complexity class: it is characterized by first-order logic with majority quantifiers on ordered finite models [3] in Descriptive Complexity Theory, or by acceptance in polynomial time on so-called Threshold Turing Machines [2], or by the machine-independent characterization below, which is most convenient for our purposes. Whenever we speak of TC^0 in the following without further qualification, we mean DLogTime-uniform TC^0.

For a complexity class C, the class $C/poly$ is defined as follows: A predicate $A(x)$ is in $C/poly$ if there is a predicate $B(x,y) \in C$ and a polynomially bounded *advice function*, i.e., a function f such that $|f(n)| \leq p(n)$ for some polynomial $p(n)$, and for which it holds that

$$\forall x\, A(x) \leftrightarrow B(x, f(|x|)) .$$

Advice functions are used to inject non-uniformity into uniform complexity classes. For example, it is well-known that $P/poly$ is equal to the class of predicates computable by non-uniform circuits of polynomial size. Analogously we have the following:

Proposition 1. *$TC^0/poly$ is the same as non-uniform TC^0.*

Proof (Sketch). For each d, there is an interpreter in TC^0 that takes as inputs a threshold circuit C of depth d and an input a to C, and outputs the value computed by C on input a. Let a non-uniform threshold circuit family $\langle C_n ; n \in \mathbb{N} \rangle$ of depth d and size $O(p(n))$ computing $A(x)$ be given. Then $A(x) \in TC^0/poly$ is seen as follows: $B(x,y)$ is the interpreter for threshold circuits of depth d, and the advice $f(n)$ is an encoding of the circuit C_n. Obviously $B(x, f(|x|))$ is equivalent to $A(x)$.

On the other hand, let $A(x) \in TC^0/poly$ given by predicate $B(x,y)$ and advice function f. Then a circuit computing $A(x)$ for inputs x of length n is constructed from the circuit computing $B(x,y)$ for inputs x of length n and y of length $|f(n)|$, by plugging into y constant subcircuits computing the bits of $f(n)$. □

Next we give the machine-independent characterization of TC^0 mentioned above:

Definition 1. *Suppose $h_0(n,\boldsymbol{x})$, $h_1(n,\boldsymbol{x}) \leq 1$. A function f is defined by concatenation recursion on notation (CRN) from g, h_0, and h_1 if*

$$\begin{aligned} f(0,\boldsymbol{x}) &= g(\boldsymbol{x}) \\ f(2n,\boldsymbol{x}) &= 2 \cdot f(n,\boldsymbol{x}) + h_0(n,\boldsymbol{x}), \textit{ provided } n \neq 0 \\ f(2n+1,\boldsymbol{x}) &= 2 \cdot f(n,\boldsymbol{x}) + h_1(n,\boldsymbol{x}) \end{aligned}$$

Let $i_k^n(x_1,\dots x_n) := x_k$, $s_0(x) := 2x$, $s_1(x) = 2x+1$, $|x| := \lceil \log_2(x+1) \rceil$, $x\#y := 2^{|x|\cdot|y|}$ and $Bit(x,i) := \lfloor \frac{x}{2^i} \rfloor \bmod 2$. The following characterization of the number-theoretic functions in TC^0 was given in [7]:

Proposition 2. *The class TC^0 is the smallest class of functions that contains 0, i_k^n, s_0, s_1, multiplication $\cdot$, $\#$, $|x|$, Bit and which is closed under composition and CRN.*

3 Theories of Bounded Arithmetic

We briefly review the necessary background on Bounded Arithmetic, for more information see [4] or [10]. The language L_2 of Bounded Arithmetic comprises the usual signature of arithmetic $0, S, +, \dot{-}, \cdot, \leq$, together with function symbols for $\lfloor \frac{1}{2}x \rfloor$, $MSP(x,i) := \lfloor x/2^i \rfloor$, $|x|$ and $\#$.

A quantifier of the form $\forall x \leq t$, $\exists x \leq t$ 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_2 by *open*. The class of sharply bounded formulas is denoted Σ_0^b or Π_0^b. For $i \in \mathbb{N}$, Σ_{i+1}^b (resp. Π_{i+1}^b) is the least class containing Π_i^b (resp. Σ_i^b) and closed under conjunction, disjunction, sharply bounded quantification and bounded existential (resp. universal) quantification.

We say that a function $f(\boldsymbol{x})$ is Σ_i^b-definable in a theory T if there is a Σ_i^b-formula $A(\boldsymbol{x},y)$ and a term $t(\boldsymbol{x})$ such that

$$\begin{aligned} &\mathbb{N} \models \forall \boldsymbol{x}\, A(\boldsymbol{x}, f(\boldsymbol{x})) \\ &T \vdash \forall \boldsymbol{x}\, \exists y \leq t(\boldsymbol{x})\, A(\boldsymbol{x},y) \\ &T \vdash \forall \boldsymbol{x}, y, z\, A(\boldsymbol{x},y) \wedge A(\boldsymbol{x},z) \rightarrow y = z\ . \end{aligned}$$

$BASIC$ denotes a set of quantifier-free axioms specifying the interpretations of the function symbols of L_2. It can most conveniently be taken as the set $BASIC$ from [4] together with the axioms for MSP and $\dot{-}$ from [14].

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

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

for each $A(x) \in \Phi$, and Φ-$LLIND$ is

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

for $A(x) \in \Phi$. In general, for $m \geq 1$, Φ-$L^m IND$ is

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

for $A(x) \in \Phi$, where $|x|_1 := |x|$ and $|x|_{m+1} := |\,|x|_m\,|$.

The theory S_2^i is the theory axiomatized by the $BASIC$ axioms and Σ_i^b-$LIND$, and R_2^i is the theory given by $BASIC$ and Σ_i^b-$LLIND$.

Definition 2. *Given a term $t \in L_2$ we define a monotonic L_2-term t^* as follows: If t is constant or a variable, then $t = t^*$. If t is $f(s)$, where f is a unary function symbol, then t^* is $f(s^*)$. If t is $s_1 \circ s_2$ for $\circ$ a binary operation other than $\dot{-}$ or MSP, then t^* is $s_1^* \circ s_2^*$. Lastly, if t is $s_1 \dot{-} s_2$ or $MSP(s_1, s_2)$, then t^* is s_1^*.*

It is easily proved in $BASIC$ + *open-LIND* that t^* is monotonic, and $t \leq t^*$. The following terms will be used frequently below. Let

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

so that $LSP(x,|y|)$ returns the number consisting of the last $|y|$ bits of x, and if w codes a sequence $\langle b_1, \ldots, b_\ell \rangle$ with $|b_i| \leq |a|$ for all i, then $\beta_a(w,i) = b_i$. The code for this 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 $|a|$. If we set $bd(a,s) := 2(2a\#2s)$, then $bd(a,s)$ is thus a bound on the code for a sequence of length $|s|$ with each item bounded by a.

For a class of formulas Φ, the replacement scheme $BB\Phi$ is

$$\begin{aligned}
&\forall x \leq |s|\, \exists y \leq t(x)\, A(x,y) \rightarrow \\
&\qquad \exists w < bd(t^*(|s|), s)\, \forall x \leq |s|\, \beta_{t^*(|s|)}(w,x) \leq t(x) \wedge A(x, \beta_{t*(|s|)}(w,x))
\end{aligned}$$

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

The theory C^0_2 is defined as $BASIC + open\text{-}LIND + BB\Sigma^b_0$. The following theorem summarizes some relations between Σ^b_i-definability in the theories defined and computational complexity.

Theorem 1. – *The Σ^b_i-definable functions in S^i_2 are exactly those in $FP^{\Sigma^P_{i-1}}$, for each $i \geq 1$ [4].*
– *The Σ^b_1-definable functions in R^1_2 are exactly those in NC [1, 5].*
– *The Σ^b_1-definable functions in C^0_2 are exactly those in TC^0 [8, 9].*

The comprehension axiom for formula $A(x)$, denoted $COMP_A(a)$, is the formula

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

The Δ^b_1-comprehension rule, Δ^b_1-$COMP$, is the following inference rule

$$\frac{A(x) \leftrightarrow B(x)}{COMP_A(t)} ,$$

where $A(x)$ is Σ^b_1 and $B(x)$ is Π^b_1, and t is an arbitrary L_2-term. Note that this rule is different from the possibly stronger Δ^b_1-comprehension *axiom*

$$\forall x \, \big(A(x) \leftrightarrow B(x)\big) \rightarrow COMP_A(a) ,$$

thus it is essential that in a sequent calculus context, the rule must not have any side formulas.

Definition 3. *Let Δ^b_1-CR be the theory axiomatized by $BASIC$, open-$LIND$ and the Δ^b_1-$COMP$ rule.*

In [9], it is proved that C^0_2 proves the Δ^b_1-$COMP$ axiom, therefore Δ^b_1-CR is a subtheory of C^0_2. But we will show that C^0_2 is not much stronger:

Theorem 2. *C^0_2 is $\forall\Sigma^b_1$-conservative over Δ^b_1-CR.*

This implies immediately:

Corollary 1. *The Σ^b_1-definable functions of Δ^b_1-CR are precisely TC^0.*

Hence $S^1_2 = \Delta^b_1$-CR implies $P = TC^0$, and $R^1_2 = \Delta^b_1$-CR implies $NC = TC^0$. We will show that the connection between the theory Δ^b_1-CR and TC^0 is still tighter: the Σ^b_2-theorems of Δ^b_1-CR can be witnessed by a type of interactive TC^0-computations to be defined below. This will allow us to show that the equality of Δ^b_1-CR to either of the stronger theories S^1_2 or R^1_2 implies a further collapse of complexity classes:

Theorem 3. *If $S^1_2 = \Delta^b_1$-CR or $R^1_2 = \Delta^b_1$-CR, then NP is contained in non-uniform TC^0.*

The method could further be generalized to show that $NP \subseteq$ non-uniform TC^0 follows from Δ_1^b-$CR \vdash \Sigma_1^b$-$L^m IND$ for any $m > 0$.

The following further axiom schemes will be used below. The Σ_1^b-length-maximization scheme, Σ_1^b-$LMAX$, is the axiom

$$\exists x \leq a\, A(x) \to \exists x \leq a\, \big(A(x) \wedge \forall y \leq a\, \big(|y| > |x| \to \neg A(y)\big)\big)$$

for every Σ_1^b-formula $A(x)$. Similarly, the Σ_1^b-double-length-maximization scheme, Σ_1^b-$LLMAX$, is the axiom

$$\exists x \leq a\, A(x) \to \exists x \leq a\, \big(A(x) \wedge \forall y \leq a\, \big(||y|| > ||x|| \to \neg A(y)\big)\big)$$

for every Σ_1^b-formula $A(x)$. The following proposition is well-known.

Proposition 3. *$S_2^1 \vdash \Sigma_1^b$-$LMAX$ and $R_2^1 \vdash \Sigma_1^b$-$LLMAX$. In fact, Σ_1^b-$LMAX$ is equivalent to Σ_1^b-$LIND$ and Σ_1^b-$LLMAX$ is equivalent to Σ_1^b-$LLIND$ over $BASIC$ + open-$LIND$.*

4 Proof of Conservativity

The following two lemmas are well-known and easily proved by the method from [6]:

Lemma 1. *The Σ_0^b-predicates are computable in TC^0. In particular, the L_2-base functions are in TC^0.*

Lemma 2. *Let f be a function in TC^0. Then the function*

$$\mu j < |x|\, (f(j, x) = 0)$$

is also in TC^0.

Lemma 3. *$\lfloor |a|/|b| \rfloor$ is contained in TC^0.*

Proof. By Lemma 2 and Lemma 1 we can define

$$\lfloor |a|/|b| \rfloor := \mu n \leq |a|\, (|a| < (n+1)|b|)\,.$$

Suppose $g(n, \boldsymbol{x}) \leq t(\boldsymbol{x})$ and s, t are L_2-terms. Then a *length-sum* is a sum of the form

$$\sum_{n=0}^{|s|} g(n, \boldsymbol{x}) \cdot 2^{n \cdot |t^*|}\,.$$

Lemma 4. *TC^0 is closed under length-sums.*

Proof. Suppose we want to define the length-sum

$$f(a, x) := \sum_{n=0}^{|a|} h(n, x) 2^{n |s^*(x)|}$$

using CRN where $h(n,x) \leq s(x)$ are functions in TC^0 . We use CRN to compute the bits of f from the most significant bit to the least significant bit. The function

$$t(i,a,x) := |a| \dot{-} \lfloor |i|/|s^*(x)| \rfloor$$

allows us to determine which term in f we are computing the bits from. The function

$$p(i,x) := |s^*(x)| \dot{-} (|i| \dot{-} \lfloor |i|/|s^*(x)| \rfloor |s^*(x)|) \dot{-} 1$$

gives us the position within a term. Define the function f' by CRN in the following way:

$$f'(2i+1,x) = f'(2i,x) = 2f(i,x) + Bit(p(i,x), h(t(i,a,x),x)).$$

Then the desired f is $f'(a\#s^*(x),x)$. □

We are now ready to show the functions in TC^0 are Σ_1^b-definable in Δ_1^b-CR.

Theorem 4. *Δ_1^b-CR can Σ_1^b-define the functions in TC^0.*

Proof. The base functions symbols are obviously Σ_1^b-definable in Δ_1^b-CR, and closure under composition is straightforward. So it suffices to show the Σ_1^b-definable functions of Δ_1^b-CR are closed under CRN.

Suppose that f is defined by CRN from $g(x)$ and $h_0(n,x), h_1(n,x)$, where g, h_0, h_1 are Σ_1^b-defined in Δ_1^b-CR. Define $t(a,x)$ to be

$$\sum_{n=0}^{|a|} cond(mod2(MSP(a, |a| \dot{-} n)), h_0(n,x), h_1(n,x)) \cdot 2^n \ ,$$

then $f(a,x) = g(x) + t(a,x) \cdot 2^{|g(x)|}$. It suffices to show the length-sum $t(a,x)$ is Σ_1^b-definable, since then $f(a,x)$ will be by composition.

Notice $k(n,x,a) := cond(mod2(MSP(a,|a| \dot{-} n)), h_0(n,x), h_1(n,x))$ is Σ_1^b-defined in Δ_1^b-CR. Let $A_k(n,a,x,y)$ be its defining formula. The value y is unique and bounded by 1. Therefore $A_f(n,x,a,1)$ and $\neg A_f(n,x,a,0)$ are true iff $k(n,x,a) = 1$ so $k(n,x,a) = 1$ is a Δ_1^b-property. We want to define the sum $\sum_{n=0}^{|a|-1} k(n,x,a) \cdot 2^n$. Δ_1^b-$COMP$ on $k(n,x,a) = 1$ implies

$$(\exists w \leq s)(\forall n \leq |a|)(Bit(n,w) = 1 \leftrightarrow k(n,x,a) = 1) \ ,$$

the value w is the desired sum and it can be proven unique using bit extensionality. □

To prove the conservativity result, we formalize the witnessing proof for C_2^0 in Δ_1^b-CR. First we define a witness predicate for Σ_1^b-formulas as follows:

- If $A(\boldsymbol{a}) \in \Sigma_0^b$ then $Wit_A(w, \boldsymbol{a}) := A(\boldsymbol{a})$.
- If $A(\boldsymbol{a})$ is of the form $B \circ C$ where $\circ$ is $\wedge$ or $\vee$ then

$$Wit_A(w, \boldsymbol{a}) := ispair(w) \wedge (Wit_B((w)_1, \boldsymbol{a}) \circ Wit_C((w)_2, \boldsymbol{a}))$$

– If $A(\boldsymbol{a})$ is of the form $\exists x \leq t\, B(x, \boldsymbol{a})$ where $B(x, \boldsymbol{a}) \in \Sigma_0^b$ then

$$Wit_A(w, \boldsymbol{a}) := w \leq t \wedge B(w, \boldsymbol{a}) \ .$$

– If $A(\boldsymbol{a})$ is of the form $\exists x \leq t\, B(x, \boldsymbol{a})$ where $B(x, \boldsymbol{a}) \in \Sigma_1^b \setminus \Sigma_0^b$, then

$$Wit_A(w, \boldsymbol{a}) := ispair(w) \wedge (w)_1 \leq t \wedge Wit_B((w)_1, (w)_2, \boldsymbol{a}) \ .$$

– If $A(\boldsymbol{a})$ is of the form $\forall x \leq |s|\ B(x, \boldsymbol{a})$ where $B(x, \boldsymbol{a}) \in \Sigma_1^b \setminus \Sigma_0^b$, then we define

$$Wit_A(w, \boldsymbol{a}) := w \leq bd(t^*, s) \wedge \forall x \leq |s|\, Wit_B(\beta_{t^*}(x, w), x, \boldsymbol{a})) \ .$$

The following lemma is true for this witness predicate:

Lemma 5. *If $A(\boldsymbol{a}) \in \Sigma_1^b$, then:*

(a) Wit_A is a Σ_0^b-predicate.
(b) There is a term t_A so that Δ_1^b-$CR \vdash Wit_A(w, \boldsymbol{a}) \to w \leq t_A$.
(c) For this term t_A, Δ_1^b-$CR \vdash \exists w \leq t_A(\boldsymbol{a})\, Wit_A(w, \boldsymbol{a}) \to A(\boldsymbol{a})$.

Proof. (a) Follows from the definition of witness and since β and the pairing functions are defined by L_2-terms. (b) and (c) are easily proved by induction on the complexity of A. □

To prove the witnessing theorem, we formalize C_2^0 in a sequent calculus LKB that has special rules for the introduction of bounded quantifiers (see [4]). In this formalization, *open-$LIND$* and $BB\Sigma_0^b$ are given as inference rules, which are shown in the proof below.

Theorem 5. *Suppose*

$$C_2^0 \vdash \Gamma \implies \Delta$$

where Γ and Δ are cedents of Σ_1^b-formulas in Let $\boldsymbol{a}$ be the free variables in this sequent. Then there is a TC^0 function f which is Σ_1^b-defined in Δ_1^b-CR such that:

$$\Delta_1^b\text{-}CR \vdash Wit_{\bigwedge \Gamma}(w, \boldsymbol{a}) \to Wit_{\bigvee \Delta}(f(w, \boldsymbol{a}), \boldsymbol{a}).$$

Proof. This is proved by induction on the number of sequents in an C_2^0 proof of $\Gamma \implies \Delta$. By cut elimination, we can assume all the sequents in the proof are Σ_1^b. Most of the cases are similar to previous witnessing arguments so we only show the ($\forall$: right) case, *open-$LIND$* case and the $BB\Sigma_0^b$ case.
($\forall$:right case) Suppose we have the inference:

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

By the induction hypothesis there is a TC^0 function g such that

$$\Delta_1^b\text{-}CR \vdash Wit_{b \leq t \wedge \bigwedge \Gamma}(w, \boldsymbol{a}, b) \to Wit_{A \vee \bigvee \Delta}(g(w, \boldsymbol{a}, b), \boldsymbol{a}, b) \ .$$

By cut-elimination, $\forall x \leq t\, A(x)$ is a Σ^b_1-formula, so t must be of the form $t = |s|$. There are two case: where A is Σ^b_0 and where A is $\Sigma^b_1 \backslash \Sigma^b_0$. In the first case, let y be $\mu i \leq |s|\, \neg A(i)$ and define f to be $g(w, \boldsymbol{a}, y)$. This is in TC^0 be Lemma 1 and Lemma 2 and it is not hard to show that

$$\Delta^b_1\text{-}CR \vdash Wit_\Gamma(w, \boldsymbol{a}) \rightarrow Wit_{\forall x \leq |s|\, A \vee \bigvee \Delta}(f(w, \boldsymbol{a}), \boldsymbol{a}) .$$

In the second case, since Wit_A is a Σ^b_0-formula, its characteristic function χ_{Wit_A} is in TC^0. Let k be the function

$$k(w, \boldsymbol{a}) = \mu j \leq |s|\, [\neg Wit_A((g(w, \boldsymbol{a}, j))_1, \boldsymbol{a}, j)] .$$

Now define $f(w, \boldsymbol{a})$ from k as follows

$$f(w, \boldsymbol{a}) = \begin{cases} \sum_{j=0}^{|s|} (g(w, \boldsymbol{a}, j))_1 \cdot 2^{j \cdot |(t')^*|} & \text{if } k(w, \boldsymbol{a}) = |s| + 1 \\ (g(w, \boldsymbol{a}, k(w, \boldsymbol{a})))_2 & \text{otherwise} \end{cases} ,$$

then it holds that

$$\Delta^b_1\text{-}CR \vdash Wit_\Gamma(w, \boldsymbol{a}) \rightarrow Wit_{\forall x \leq |s|\, A \vee \bigvee \Delta}(f(w, \boldsymbol{a}), \boldsymbol{a}) .$$

(*open-LIND* **case**) Suppose we have the inference

$$\frac{A(b), \Gamma \implies A(Sb), \Delta}{A(0), \Gamma \implies A(|s|), \Delta}$$

where A is an open formula and s is a term in L_2. By the induction hypothesis there is a TC^0 function g such that

$$\Delta^b_1\text{-}CR \vdash Wit_{A(b) \wedge \bigwedge \Gamma}(w, b, \boldsymbol{a}) \rightarrow Wit_{A(Sb) \vee \bigvee \Delta}(g(w, b, \boldsymbol{a}), b, \boldsymbol{a}).$$

From our definition of the Wit predicate and Lemma 1, we know TC^0 contains the predicate $Wit_{\bigvee \Delta}$. Define

$$f(w, \boldsymbol{a}) := g(w, (\mu y < |s|\,)(\neg Wit_{\bigvee \Delta}((g(w, y, \boldsymbol{a})_2), y, \boldsymbol{a})), \boldsymbol{a}).$$

The idea is $f(w, \boldsymbol{a})$ runs g on the least value y less than $|s|$ that produces a witness for Δ. If no such value exists then it must be the case that $A(|s|)$ holds and so, as A is open, the cedent is trivially witnessed. From this it is not hard to show:

$$\Delta^b_1\text{-}CR \vdash Wit_{A(0) \wedge \bigwedge \Gamma}(w, \boldsymbol{a}) \rightarrow Wit_{A(|r|) \vee \bigvee \Delta}(f(w, \boldsymbol{a}), \boldsymbol{a}).$$

($BB\Sigma^b_0$**:case**) Suppose we have the inference:

$$\frac{\Gamma \implies \forall x \leq |s|\, \exists y \leq t\, A(x, y), \Delta}{\Gamma \implies \exists v \leq bd(t^*, s)\, \forall x \leq |s|\, (\beta_{t^*}(x, v) \leq t \wedge A(x, \beta_{t^*}(x, v))), \Delta}$$

where s, t are terms in L_2 and $A(x, y) \in \Sigma^b_0$. By the induction hypothesis there is a TC^0 function g such that

$$\Delta^b_1\text{-}CR \vdash Wit_{\bigwedge \Gamma}(w, \boldsymbol{a}, b) \rightarrow Wit_{\forall x \leq |s|\, \exists y \leq t\, A \vee \bigvee \Delta}(g(w, \boldsymbol{a}), \boldsymbol{a}) .$$

For this case, it suffices to notice that the predicates

$$Wit_{\forall x \leq |s|\, \exists y \leq t\, A}$$

and

$$Wit_{\exists v \leq bd(t^*,s)\, \forall x \leq |s|\, (\beta_{t^*}(x,v) \leq t \,\wedge\, A)}$$

are the same. Hence, if we let $f = g$ then

$$\Delta_1^b\text{-}CR \vdash Wit_\Gamma(w, \boldsymbol{a}, b) \rightarrow Wit_{\exists w \leq bd(t^*,s)\, \forall x \leq |s|\, A} \vee \bigvee{}_\Delta(f(w, \boldsymbol{a}), \boldsymbol{a}).$$

This completes the cases and the proof. □

Now Thm. 2 follows from this witnessing theorem as follows: Suppose C_2^0 proves a Σ_1^b-formula $A(\boldsymbol{x})$. Then by Theorem 5, taking Γ to be the empty cedent, Δ_1^b-$CR \vdash Wit_A(g(\boldsymbol{x}), \boldsymbol{x})$, where g is a TC^0 function. By Lemma 5, we have Δ_1^b-$CR \vdash A(\boldsymbol{x})$. □

5 Counterexample Computations with TC^0 functions

In this section we view binary relations $R(x, y)$ in TC^0 as *optimization problems*: given x, the task is to find a solution y of maximal length $|y| \leq |x|$ such that $R(x, y)$ holds. We consider a particular way of solving such optimization problems, viz. *counterexample computations* as introduced implicitly in [12] and made explicit in [13, 11].

A counterexample computation is performed by two agents: Student, who has limited computational power, and Teacher who has unlimited knowledge. In order to find a maximal solution, Student can ask questions of the form "Is y a maximal solution?", to which Teacher can either reply "yes" or provide a counterexample, i.e., a better solution.

There are two natural measures of complexity for counterexample computations: the computational power of Student, and the number of counterexamples. Note that every optimization problem can be solved with $O(|x|)$ many counterexamples by the trivial Student, who just repeats each counterexample provided as his next question.

Here we are interested in the case where Student has the computational capabilities of TC^0 and the number of counterexamples is bounded by a constant. We will show that the hypothesis that every optimization problem in TC^0 can be computed in this way, formalized by principle $\Omega(TC^0)$ below, implies that every NP predicate computable by non-uniform TC^0 circuits.

For an optimization problem $R(x, y)$ let $R^*(x, y, z)$ be defined by

$$|y| \leq |x| \wedge \big(y > 0 \rightarrow R(x, y)\big) \wedge \big(|y| < |z| \leq |x| \rightarrow \neg R(x, z)\big) \ ,$$

so that $\forall z\, R^*(x, y, z)$ expresses that $y = 0$ or y is a maximal solution.

Principle $\Omega(TC^0)$: for every predicate $R(x, y) \in TC^0$ there are $k \in \mathbb{N}$ and functions $f_1, \ldots f_k \in TC^0$, such that

Either	$\forall z\, R^*(a, f_1(a), z)$	or if b_1 is such that $\neg R^*(a, f_1(a), b_1)$,
then either	$\forall z\, R^*(a, f_2(a, b_1), z)$	or if b_2 is such that $\neg R^*(a, f_2(a, b_1), b_2)$,
	$\vdots$	
then	$\forall z\, R^*(a, f_k(a, b_1, \dots, b_{k-1}), z)$.	

Proposition 4. *$\Omega(TC^0)$ implies $NP \subseteq$ non-uniform TC^0.*

Proof. Let A be NP-complete under TC^0-reductions, and be given by $x \in A \leftrightarrow \exists w \le x\ B(x, w)$ with $B \in TC^0$. We say that w witnesses x if $w \le x \wedge B(x, w)$ holds.

We will construct an advice function h with $|h(n)| \le n^{O(1)}$ and $g \in TC^0$ such that $g(x, h(|x|)$ witnesses x for all $x \in A$, i.e.,

$$x \in A \text{ iff } B(x, g(x, h(|x|))) \ , \tag{1}$$

and hence $A \in TC^0/poly$, assuming $\Omega(TC^0)$.

Let the relation $R(a, b)$ be defined by

a and b code sequences, and $length(a) \ge length(b)$
and for all $i \le length(b)$: $(b)_i$ witnesses $(a)_i$.

Obviously $R \in TC^0$, so by $\Omega(TC^0)$ there are functions $f_1, \dots, f_k$ that for a sequence $a = \langle x_1, \dots, x_m \rangle$ interactively compute a maximal sequence b of witnesses for an initial segment of a.

For a fixed length n, let $V_1 := \{\, x \in A \ ;\ |x| = n \,\}$, and for each $x \in V_1$, let $w(x)$ be a canonical witness. Algorithm W below computes a pair $\langle j, w \rangle$ from an input $a = \langle x_1, \dots, x_k \rangle \in V_1^k$ such that w witnesses x_j. Since we assume $\Omega(TC^0)$, Algorithm W always halts at one of the **stop** instructions.

```
y := f_1(a)
if length(y) ≥ 1 and R(a, y) then
    output ⟨1, (y)_1⟩
    stop
fi
for j from 2 to k do
    y := f_j(a, b_1, ..., b_{j-1})
    if length(y) ≥ j and R(a, y) then
        output ⟨j, (y)_j⟩
        stop
    fi
od
```

Algorithm W. b_j is defined as $\langle w(x_1), \dots, w(x_{j-1}) \rangle$.

For a set $Q \subseteq V_1$ with $|Q| = k - 1$ and $v \in V_1 \setminus Q$ we define Q *helps* v if for some ordering $a := \langle x_1, \dots, x_{j-1}, v, x_{j+1}, \dots, x_k \rangle$ of $Q \cup \{v\}$, Algorithm W on input a outputs a pair $\langle j, w \rangle$ such that w witnesses v.

As there is only a constant number $k!$ of orderings of $Q \cup \{v\}$, there is a function in TC^0 that, given Q, v and canonical witnesses for the elements of Q, uses Algorithm W to decide whether Q helps v, and if so computes a witness $w(Q, v)$ for v.

There are at least $\binom{|V_1|}{k}$ pairs $\langle Q, v\rangle$ such that Q helps v, but there are only $\binom{|V_1|}{k-1}$ possible sets Q of size $k-1$. Hence there is a set $Q_1 \subseteq V_1$ such that Q_1 helps at least $\frac{|V_1|-k+1}{k}$ different elements of V_1.

Inductively we define $V_{i+1} := \{\, v \in V_i \,;\, Q_i \text{ does not help } v \,\}$, and by the same argument as above, if $|V_{i+1}| > k$ then there is a set $Q_{i+1} \subseteq V_{i+1}$ that helps at least $\frac{|V_{i+1}|-k+1}{k}$ elements of $V_{i+1} \setminus Q_{i+1}$.

Let t be the least i such that $|V_t| \leq k$, then since $|V_{i+1}| < \left(\frac{k-1}{k}\right)^i |V_1| + k$ we get $t = \lceil \log_{k/(k-1)} |V_1| \rceil = O(n)$. For $i < t$ let S_i be the sequence of pairs $\langle x, w(x)\rangle$ for $x \in Q_i$, and let S_t be the sequence of pairs $\langle x, w(x)\rangle$ for $x \in V_t$. Finally, let the advice $h(n)$ be $S := \langle S_1, \ldots, S_t\rangle$. Note that $|S| = O(kn^2)$.

Finally, Algorithm G computes a witness for $v \in V_1$ from inputs v and S. By the remark above, lines 5–6 of Algorithm G can be implemented in TC^0,

```
if v occurs in S then
    output w(v)        (* also occurs in S next to v *)
else
    for j ∈ {1, ..., t − 1} do in parallel
        if Q_j helps v then
            w_j := w(Q_j, v)
    od
    output w_j with j < t minimal
fi
```

Algorithm G.

and hence the function g computed by Algorithm G is in TC^0. By construction $g(x, h(|x|))$ witnesses x iff there is a witness for x, hence the equivalence (1) holds. ◻

We now consider a variant where the measure to be maximized is $||y||$ instead of $|y|$. Principle $\Omega^*(TC^0)$ is thus exactly the same as $\Omega(TC^0)$, only with the relation $R^*(x, y, z)$ replaced by $R^{**}(x, y, z)$, which is defined as

$$||y|| \leq ||x|| \wedge \big(y > 0 \rightarrow R(x, y)\big) \wedge \big(||y|| < ||z|| \leq ||x|| \rightarrow \neg R(x, z)\big) \ .$$

Proposition 5. *$\Omega^*(TC^0)$ implies $NP \subseteq$ non-uniform TC^0.*

Proof. Modify the proof of Prop. 4 as follows: Let $\ell := 2^{k-1}$. Algorithm W is replaced by Algorithm W*, which gets input $a = \langle x_1, \ldots, x_\ell\rangle \in V_1^\ell$.

For $Q \subseteq V_1$ with $|Q| = \ell - 1$ and $v \in V_1 \setminus Q$, define Q *helps* v if for some ordering $a := \langle x_1, \ldots, x_{m-1}, v, x_{m+1}, \ldots, x_\ell\rangle$ of $Q \cup \{v\}$, Algorithm W* on

```
y := f_1(a)
if length(y) ≥ 1 and R(a, y) then
    output ⟨1, (y)_1⟩
    stop
fi
for j from 2 to k do
    y := f_j(a, b_1, ..., b_{j−1})
    if length(y) ≥ 2^{j−1} and R(a, y) then
        w := ⟨(y)_{2^{j−2}+1}, ..., (y)_{2^{j−1}}⟩
        output ⟨j, w⟩
        stop
    fi
od
```

Algorithm $\mathbf{W}^*$. b_j is defined as $\langle w(x_1), \dots, w(x_{2^{j-1}})\rangle$.

input a outputs a pair $\langle j, w\rangle$ such that either $j = m = 1$ and w witnesses v, or $2^{j-2} < m \leq 2^{j-1}$ and w is a sequence of length 2^{j-2} such that $(w)_{m-2^{j-2}}$ witnesses v.

The definition of the advice S is as before, only with k replaced by ℓ everywhere. So Algorithm G on input v and S will still output a witness for v if there is one. □

6 KPT witnessing for Δ^b_1-CR

In [12] it was shown that the $\exists\forall\Sigma^b_{i+1}$-theorems of T^i_2 can be witnessed by counterexample computations using $FP^{\Sigma^P_i}$-functions and constantly many counterexamples. For this to be true for $i = 0$, T^0_2 needs to be defined as having function symbols for all functions in FP.

Analogously, we now show that the $\exists\forall\Delta^b_1$-theorems of Δ^b_1-CR can be witnessed by counterexample computations using TC^0-functions and constantly many counterexamples. This will be the main tool for proving Thm. 3, but the witnessing theorem and its proof might be of independent interest.

Theorem 6. *Assume Δ^b_1-$CR \vdash \exists x\, \forall y\, A(a, x, y)$, where A is Δ^b_1 w.r.t. Δ^b_1-CR. Then there are $k \in \mathbb{N}$ and functions $f_1, \dots, f_k \in TC^0$, that are Σ^b_1-definable in Δ^b_1-CR, s.t. Δ^b_1-CR proves*

$$A(a, f_1(a), b_1) \vee A(a, f_2(a, b_1), b_2) \vee \dots \vee A(a, f_k(a, b_1, \dots, b_{k-1}), b_k) \ .$$

Proof. Let $\{ f_n \ ; \ n \geq 1 \}$ be an enumeration of all functions in TC^0 s.t. f_n is n-ary and every function in TC^0 occurs in the list infinitely often (possibly with dummy arguments). Assume that A is Δ^b_1 w.r.t. Δ^b_1-CR and Δ^b_1-$CR \vdash \exists x\, \forall y\, A(a, x, y)$, but the conclusion of the theorem does not hold. Then by compactness there is a model

$$M \models \Delta^b_1\text{-}CR + \{\neg A(c, f_1(c), d_1), \dots, \neg A(c, f_n(c, d_1, \dots, d_{n-1}), d_n), \dots\}$$

for new constants $c, d_1, d_2, \ldots$

Define $M^* := \{f_1(c), f_2(c, d_1), \ldots, f_n(c, d_1, \ldots, d_{n-1}), \ldots\}$. By the construction of the enumeration f_n, $\mathbb{N} \cup \{c, d_1, d_2, \ldots\} \subseteq M^*$, and M^* is closed under all functions in TC^0.

We first show $M^* \preceq_{\Sigma^b_0} M$, i.e., for every Σ^b_0-formula $B(\boldsymbol{x})$ and all parameters $\boldsymbol{a} \in M^*$,

$$M \models B(\boldsymbol{a}) \text{ iff } M^* \models B(\boldsymbol{a}) \ .$$

This is proved by induction on the complexity of $B(\boldsymbol{x})$. The only interesting case is to show that for $B(\boldsymbol{x}) = \exists y \leq |t(\boldsymbol{x})|\, A(\boldsymbol{x}, y)$, $M \models B(\boldsymbol{a})$ implies $M^* \models B(\boldsymbol{a})$. Consider the function $f(\boldsymbol{x}) = \mu y \leq |t(\boldsymbol{x})|\, A(\boldsymbol{x}, y)$. This function is in TC^0, hence $f(\boldsymbol{a}) \in M^*$, and if $M \models B(\boldsymbol{a})$, then $M \models A(\boldsymbol{a}, f(\boldsymbol{a}))$, therefore $M^* \models A(\boldsymbol{a}, f(\boldsymbol{a}))$ holds by the induction hypothesis.

Hence if $A(\boldsymbol{x})$ is Π^b_1 and $B(\boldsymbol{x})$ is Σ^b_1 and $\boldsymbol{a} \in M^*$, then $M \models A(\boldsymbol{a})$ implies $M^* \models A(\boldsymbol{a})$ and $M^* \models B(\boldsymbol{a})$ implies $M \models B(\boldsymbol{a})$.

Let $\Delta^b_1\text{-}CR_0$ denote $BASIC{+}\mathit{open}\text{-}LIND$, and inductively define $\Delta^b_1\text{-}CR_{i+1}$ to be the closure of $\Delta^b_1\text{-}CR_i$ under unnested applications of $\Delta^b_1\text{-}COMP$, and Γ_i to be the set of formulas that are Δ^b_1 w.r.t. $\Delta^b_1\text{-}CR_i$. Hence $\Delta^b_1\text{-}CR_{i+1}$ is axiomatized by all theorems of $\Delta^b_1\text{-}CR_i$ and the axioms $COMP_A$ for all formulas $A \in \Gamma_i$, $\Delta^b_1\text{-}CR = \bigcup_i \Delta^b_1\text{-}CR_i$ and the set of formulas that are Δ^b_1 w.r.t. $\Delta^b_1\text{-}CR$ is $\Gamma := \bigcup_i \Gamma_i$.

We shall show by simultaneous induction that for all i, $M^* \models \Delta^b_1\text{-}CR_i$ and $M^* \preceq_{\Gamma_i} M$. Obviously $M^* \models BASIC$. Now let $M^* \models B(0) \wedge \neg B(|a|)$ for some open formula $B(x)$ and $a \in M^*$. Then also $M \models B(0) \wedge \neg B(|a|)$, hence there is a least $b \in M$ such that $M \models b < |a| \wedge B(b) \wedge \neg B(b+1)$. Since the function $f(x) := \mu y < |x| \neg B(y+1)$ is in TC^0, $f(a) = b \in M^*$, and $M^* \models B(b) \wedge \neg B(b+1)$. This shows $M^* \models \mathit{open}\text{-}LIND$ and thus $M^* \models \Delta^b_1\text{-}CR_0$.

Now assume that $M^* \models \Delta^b_1\text{-}CR_i$, and let $B(\boldsymbol{x}) \in \Gamma_i$. This means there are a Σ^b_1-formula $B^\Sigma(\boldsymbol{x})$ and a Π^b_1-formula $B^\Pi(\boldsymbol{x})$ such that

$$\Delta^b_1\text{-}CR_i \vdash B^\Sigma(\boldsymbol{x}) \leftrightarrow B(\boldsymbol{x}) \leftrightarrow B^\Pi(\boldsymbol{x}) \ .$$

Let $\boldsymbol{a} \in M^*$, then we have

$$M \models B(\boldsymbol{a}) \Longrightarrow M \models B^\Pi(\boldsymbol{a}) \stackrel{(\dagger)}{\Longrightarrow} M^* \models B^\Pi(\boldsymbol{a}) \stackrel{(*)}{\Longrightarrow} M^* \models B(\boldsymbol{a})$$

$$M^* \models B(\boldsymbol{a}) \stackrel{(*)}{\Longrightarrow} M^* \models B^\Sigma(\boldsymbol{a}) \stackrel{(\dagger)}{\Longrightarrow} M \models B^\Sigma(\boldsymbol{a}) \Longrightarrow M \models B(\boldsymbol{a})$$

The implications marked $(*)$ hold since $M^* \models \Delta^b_1\text{-}CR_i$, and those marked $(\dagger)$ hold by $M^* \preceq_{\Sigma^b_0} M$. Hence we have shown $M^* \preceq_{\Gamma_i} M$.

Again, let $B(x) \in \Gamma_i$, and $a \in M^*$. Then the characteristic function of B, χ_B, is in TC^0, and from it we can define a function f_B using CRN that satisfies

$$M \models \forall i < |a| \left(Bit(f_B(a), i) = 1 \leftrightarrow \chi_B(i) = 1 \right) .$$

Since $\chi_B(i) = 1$ is in Γ_i, this formula is also in Γ_i, and hence it also holds in M^*, and furthermore

$$M^* \models \forall i < |a| \left(\chi_B(i) = 1 \leftrightarrow B(i) \right) ,$$

since this formula is in Γ_i and holds in M. Hence $M^* \models COMP_B$, and we have shown that $M^* \models \Delta_1^b\text{-}CR_{i+1}$.

By induction, $M^* \models \Delta_1^b\text{-}CR$ and $M^* \preceq_\Gamma M$. Finally, we show that

$$M^* \models \forall x\, \exists y\, \neg A(c, x, y) ,$$

which contradicts the assumption that $\Delta_1^b\text{-}CR \vdash \exists x \forall y A(a, x, y)$, and thus proves the theorem. Indeed, for $a = f_n(c, d_1, \ldots, d_{n-1}) \in M^*$, let $b = d_n$, then by construction $M \models \neg A(c, a, b)$, and since $M^* \preceq_\Gamma M$, also $M^* \models \neg A(c, a, b)$. □

Note that the proof does not show that M^* satisfies the Δ_1^b-comprehension *axiom*, but only the $\Delta_1^b\text{-}COMP$ rule.

Corollary 2. *If $S_2^1 = \Delta_1^b\text{-}CR$, then $\Omega(TC^0)$ holds, and $R_2^1 = \Delta_1^b\text{-}CR$ implies $\Omega^*(TC^0)$.*

Proof. Let $R(x, y)$ be a predicate in TC^0, then $R(x, y)$ is Δ_1^b w.r.t. $\Delta_1^b\text{-}CR$, and hence also $R^*(x, y, z)$ and $R^{**}(x, y, z)$ are Δ_1^b w.r.t. $\Delta_1^b\text{-}CR$. Now we have

$$\begin{array}{ll} S_2^1 \vdash \exists y\, \forall z\, R^*(a, y, z) & \text{by } \Sigma_1^b\text{-}LMAX \\ R_2^1 \vdash \exists y\, \forall z\, R^{**}(a, y, z) & \text{by } \Sigma_1^b\text{-}LLMAX \end{array}$$

and thus if $S_2^1 = \Delta_1^b\text{-}CR$, then $\Delta_1^b\text{-}CR \vdash \exists y\, \forall z\, R^*(a, y, z)$, and by Thm. 6 there are $k \in \mathbb{N}$ and functions $f_1, \ldots f_k \in TC^0$ such that

$$R^*(a, f_1(a), b_1) \vee R^*(a, f_2(a, b_1), b_2) \vee \ldots \vee R^*(a, f_k(a, b_1, \ldots, b_{k-1}), b_k) ,$$

i.e., principle $\Omega(TC^0)$ holds. By the same argument with R^{**} instead of R^*, if $R_2^1 = \Delta_1^b\text{-}CR$ then $\Omega^*(TC^0)$ holds. □

Corollary 2 together with Prop. 4 and 5 prove Thm. 3. The proof of Thm. 6 suggests some open question:

- First, is $\Delta_1^b\text{-}CR = \Delta_1^b\text{-}CR_i$ for some i?
- For $f \in TC^0$, is there a relationship between the minimal i s.t. f is Σ_1^b-definable in $\Delta_1^b\text{-}CR_i$ and the nesting depth of CRN required to define f in the function algebra? Note that the proof of Thm. 4 actually shows every function in TC^0 that can be defined by i nested applications of CRN is Σ_1^b-definable in $\Delta_1^b\text{-}CR_i$.
- Moreover, is there a relation between either of these complexity measures and the depth of a TC^0 circuit family computing f?

References

1. B. Allen. Arithmetizing uniform *NC*. *Annals of Pure and Applied Logic*, 53:1–50, 1991.
2. E. Allender. The permanent requires large uniform threshold circuits. To appear in *Chicago Journal of Theoretical Computer Science*. Preliminary Version appeared in COCOON'96, 1998.

3. D. A. M. Barrington, N. Immermann, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.
4. S. R. Buss. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.
5. P. Clote. A first order theory for the parallel complexity class NC. Technical Report BCCS-89-01, Boston College, January 1989.
6. P. Clote. On polynomial size Frege proofs of certain combinatorial principles. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, volume 23 of *Oxford Logic Guides*, pages 162–184. Clarendon Press, Oxford, 1993.
7. 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.
8. 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.
9. J. Johannsen and C. Pollett. On proofs about threshold circuits and counting hierarchies (extended abstract). In *Proc. 13th LICS*, pages 444–452, 1998.
10. J. Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory.* Cambridge University Press, 1995.
11. J. Krajíček, P. Pudlák, and J. Sgall. Interactive computations of optimal solutions. In B. Rovan, editor, *Mathematical Foundations of Computer Science*, pages 48–60. Springer, 1990.
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. P. Pudlák. Some relations between subsystems of arithmetic and complexity of computations. In Y. N. Moschovakis, editor, *Logic from Computer Science*, pages 499–519. Springer, New York, 1992.
14. 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.