Theories for $\mathbf{TC}^0$ and other Small Complexity Classes

Phuong Nguyen and Stephen Cook

November 22, 2004

Abstract

We present a general method for introducing finitely axiomatizable "minimal" second-order theories for various subclasses of $\mathbf{P}$. We show that our theory $\mathbf{VTC}^0$ for the complexity class $\mathbf{TC}^0$ is RSUV isomorphic to the first-order theory $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ of Johannsen and Pollett, thus showing that the latter theory is finitely axiomatizable.

1 Introduction

Non-uniform $\mathbf{AC}^0$ is the class of languages accepted by polynomial-size families of constant-depth Boolean circuits (where the gates have unbounded fan-in). Non-uniform $\mathbf{TC}^0$ is defined similarly, where the circuits may contain *majority* gates (i.e., gates with unbounded fan-in, which output 1 if and only if the number of 1 inputs is more than the number of 0 inputs), and for non-uniform $\mathbf{AC}^0(m)$[1] the additional gates are mod m gates, i.e., gates with unbounded fan-in which output 1 if and only if the number of 1 inputs is exactly 1 modulo m.

Each of these classes has a uniform version, where the families of circuits are uniform. Here we consider $\mathbf{FO}$–uniformity [17], i.e., each circuit family can be described by some first–order formula. We will focus on the uniform classes, and will simply use $\mathbf{AC}^0$, $\mathbf{TC}^0$ and $\mathbf{AC}^0(m)$ without the adjective "uniform".

Each of these classes can be defined more generally as a class of *relations* rather than languages. A class $\mathbf{C}$ is then associated with a *function* class $\mathbf{FC}$, which is essentially the set of functions of at most polynomial growth whose *bit graphs* are in $\mathbf{C}$. Then $\mathbf{TC}^0$ (resp. $\mathbf{FTC}^0$) is the class of problems (resp. functions) $\mathbf{AC}^0$ reducible to the *counting function*, which outputs the number of 1 bits in the input string. The same holds for $\mathbf{AC}^0(m)$ and $\mathbf{FAC}^0(m)$, with the *modulo m* function instead of the counting function.

It is known that
$$\mathbf{AC}^0 \subsetneq \mathbf{AC}^0(p) \subsetneq \mathbf{AC}^0(pq) \subseteq \mathbf{ACC} \subseteq \mathbf{TC}^0 \subseteq \mathbf{NC}^1,$$
for any distinct prime numbers p, q, where $\mathbf{ACC} = \bigcup_{m=2}^{\infty} \mathbf{AC}^0(m)$. However it is an open question whether any of last three inclusions is strict. It is also unknown, for example, whether $\mathbf{AC}^0(6) \subsetneq \mathbf{NC}^1$, although $\mathbf{AC}^0(p) \neq \mathbf{AC}^0(q)$ for distinct prime numbers p, q.

In this paper we study second-order logical theories associated with these and other complexity classes. We show that our theories $\mathbf{VTC}^0$ and $\mathbf{V}^0(m)$ characterize $\mathbf{TC}^0$ and $\mathbf{AC}^0(m)$ in the same way that Buss's theories $\mathbf{S}_2^1, \mathbf{S}_2^2, \ldots$ characterize the polynomial time hierarchy [4]. Thus we show that $\mathbf{FTC}^0$ is precisely the class of $\mathbf{\Sigma}_1^1$-definable functions of $\mathbf{VTC}^0$, and similarly $\mathbf{FAC}^0(m)$ is the class of $\mathbf{\Sigma}_1^1$-definable functions of $\mathbf{V}^0(m)$.

In Section 4 we show that our theory $\mathbf{VTC}^0$ is *RSUV isomorphic* to $\mathbf{\Delta}_1^b$-$\mathbf{CR}$, a "minimal" first–order theory that also characterizes $\mathbf{TC}^0$ [21] but which is defined very differently from $\mathbf{VTC}^0$. Since $\mathbf{VTC}^0$ is finitely

[1] $\mathbf{AC}^0(2)$ is also called $\mathbf{ACC}(2)$ in [19]

axiomatizable, it follows that $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ is also, and this answers an open question in [21] by showing that there is a constant upper bound to the nesting depth of the $\mathbf{\Delta}_1^b$ bit-comprehension rule required to prove theorems in $\mathbf{\Delta}_1^b$-$\mathbf{CR}$. Our RSUV isomorphism is more difficult than the original ones given in [28, 29], as we explain below in Section 1.2.

The theory $\mathbf{VTC}^0$ is obtained by adding to the "base" theory $\mathbf{V}^0$ [30, 11] (a theory that characterizes $\mathbf{AC}^0$) the axiom *NUMONES* which encodes the counting function which is complete for $\mathbf{TC}^0$. This is indeed a generic method that can be used to develop "minimal", finitely axiomatizable theories characterizing other small classes, including the sequence

$$\mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{SL} \subseteq \mathbf{NL} \subseteq \mathbf{NC} \subseteq \mathbf{P}$$

In proving that our theories characterize the corresponding classes, we follow the approach laid down in [11] which uses "minimal", universal theories over the languages of the functions in the corresponding classes. The universal counterpart of $\mathbf{VTC}^0$ is called $\overline{\mathbf{VTC}}^0$. The main tasks remaining are to (i) show that the universal theories (such as $\overline{\mathbf{VTC}}^0$) are conservative extensions of the original theories (e.g., $\mathbf{VTC}^0$); and (ii) prove the *Witnessing Theorems* for the universal theories. The general results in Theorem 3.15 and Theorem 3.20 (the General Witnessing Theorem) should be useful for these purposes in other contexts.

This paper is based in part on its precursors [25] and [26].

1.1 Previous First-Order Theories for our Classes

In [9], Clote and Takeuti introduce the notion of *essentially sharply bounded* (esb) formulas in a theory $\mathcal{T}$. They introduce the first-order theories $\mathbf{TAC}^0(2)$, $\mathbf{TAC}^0(6)$, and $\mathbf{TTC}^0$, and show that a function is esb-definable in one of these theories iff it is in $\mathbf{AC}^0(2)$, $\mathbf{AC}^0(6)$, or $\mathbf{TC}^0$, respectively. However, the notion of esb-definable seems unnecessarily complicated.

In [18], Johannsen introduces the first-order theory $\overline{\mathbf{R}}^0$, and shows that the class of $\mathbf{TC}^0$ functions is exactly the class of functions Σ_1^b-definable in $\overline{\mathbf{R}}^0$. In [20], Johannsen and Pollett introduce a hierarchy $\{\mathbf{C}_k^0\}_{k\geq 1}$ of first-order theories, where $\mathbf{C}_k^0$ characterizes the class of functions computable by families of constant-depth threshold circuits of size bounded by $\tau_k(n)$, where $\tau_1(n) = O(n), \tau_{k+1}(n) = 2^{\tau_k(\log n)}$. In particular, $\mathbf{C}_2^0$ captures $\mathbf{TC}^0$. Later Johannsen and Pollett [21] introduce the "minimal" theory $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ for $\mathbf{TC}^0$ mentioned above. This theory is defined using a set of axioms (**BASIC** together with **Open-LIND**), and the Δ_1^b *bit-comprehension rule*. It is easy to see that $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ is a subset of both $\overline{\mathbf{R}}^0$ and $\mathbf{C}_2^0$, and it follows from a result of Cook and Thapen [15] that $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ is a proper subset of both unless the RSA encryption scheme can be cracked in polynomial time.

The equational theories $A2V$ and TV introduced by Johannsen [19] characterize $\mathbf{AC}^0(2)$ and $\mathbf{TC}^0$, respectively. These theories appear to be RSUV isomorphic respectively to our second-order theories $\mathbf{V}^0(2)$ and $\mathbf{VTC}^0$. One direction is clear: the axioms of the equational theories translate to theorems of the second-order theories. To show the reverse direction would require working out detailed proofs in the equational theories.

We show in Section 4 that $\mathbf{VTC}^0$ is RSUV isomorphic to $\mathbf{\Delta}_1^b$-$\mathbf{CR}$. From this and the previous paragraph it appears that TV and $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ are equivalent.

1.2 Second-Order Theories for $\mathbf{TC}^0$

In [20], the first-order theories $\mathbf{C}_{k+1}^0$ ($k \geq 1$) is shown to be RSUV isomorphic to the second-order theories $\mathbf{D}_k^0$. Thus, $\mathbf{D}_1^0$ can be seen as a theory for $\mathbf{TC}^0$. By the results of Cook and Thapen [15] discussed above, $\mathbf{D}_1^0$ appears to be stronger than our theory $\mathbf{VTC}^0$.

In [24], Krajíček introduces the theory $(I\Sigma_0^{1,b})^{count}$ and notes that it should correspond to constant-depth $\mathbf{FC}$, where $\mathbf{FC}$ is an extension of Frege proof systems. It turns out that our theory $\mathbf{VTC}^0$ is essentially the same as $(I\Sigma_0^{1,b})^{count}$, but we note that Krajíček does not treat his theory in detail.

As argued in [11], it seems that the second-order logic used here is more appropriate for reasoning about small complexity classes. The usual first-order theories of bounded arithmetic (including most of those described above for $\mathbf{TC}^0$) include multiplication as a primitive operation, and include axioms such as $x \cdot y = y \cdot x$. Our second-order theories have no primitive operations on second-order objects (strings) other than length. One advantage of this simplicity comes in the easy description of the propositional translations of these theories [11]. In order to show the RSUV isomorphism in Section 4 between $\mathbf{VTC}^0$ and $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ we must define binary multiplication in $\mathbf{VTC}^0$ and prove its properties, which is not an easy task. But the alternative of simply assuming the commutative and distributive laws as axioms is "cheating", rather like throwing in axioms for the commutativity of multiplication in a propositional proof system for $\mathbf{TC}^0$.

1.3 Organization

Section 2 presents the syntax and semantics of our second-order theories, and defines the second-order versions of the complexity classes $\mathbf{AC}^0$, $\mathbf{TC}^0$, $\mathbf{AC}^0(m)$, and $\mathbf{ACC}$. Characterizations of $\mathbf{TC}^0$ and $\mathbf{AC}^0(m)$ are given in terms of threshold quantifiers and modulo m quantifiers, respectively.

Section 3 defines a finitely axiomatized theory for each of the complexity classes mentioned in the introduction, and introduces a universal conservative extension of each of these theories which has function symbols for each function in the associated class. The main theorems state that the Σ_1^B-definable functions in each theory are the functions in the associated complexity class. The theory $\mathbf{VTC}^0$ for $\mathbf{TC}^0$ is treated in detail, and then a general method is introduced for defining theories for other subclasses of $\mathbf{P}$. A general witnessing theorem is proved.

Section 4 proves that our finitely-axiomatized second-order theory $\mathbf{VTC}^0$ is isomorphic to the first-order theory $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ of Johannsen and Pollett. It follows that there is a fixed upper bound of the nesting depth of applications of the Δ_1^b bit-comprehension rule required for proofs in $\mathbf{\Delta}_1^b$-$\mathbf{CR}$, which answers an open question in [21].

Section 5 summarizes our main contributions. Appendix A gives some details of the RSUV isomorphism proof, and Appendix B shows how the proof of the Pigeonhole Principle can be formalized in $\mathbf{VTC}^0$.

2 Second-Order Logic

2.1 Syntax and Semantics

We use the two-sorted syntax of Zambella [30, 31] (see also [11, 10]), which was inspired by Buss's second-order theories defined in [4]. Our language has two sorts of variables: the number variables $x, y, z, \ldots$ whose intended values are natural numbers; and string variables $X, Y, Z, \ldots$, whose intended values are finite sets of natural numbers (which represent binary strings). Our two-sorted vocabulary $\mathcal{L}_A^2$ extends that of Peano Arithmetic:

$$\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =^1, =^2].$$

Here $|\ |$ is a function from strings to numbers, and the intended meaning of $|X|$ is 1 plus the largest element of X. The binary predicate $\in$ denotes set membership. We will use the abbreviation $X(t)$ for $t \in X$. The equality predicates $=^1$ and $=^2$ are for numbers and strings, respectively. We will write $=$ for both $=^1$ and $=^2$; the exact meaning will be clear from the context. The other symbols have their standard meanings.

Number terms are built from the constants 0,1, variables $x, y, z, \ldots$, and length terms $|X|$ using $+$ and $\cdot$. We use $s, t, \ldots$ for number terms. The only *string terms* are string variables $X, Y, Z, \ldots$. The atomic formulas are $\top$, $\bot$, (for True, False), $s = t$, $X = Y$, $s \le t$, $t \in X$ for any number terms s, t and string variables X, Y. Formulas are built from atomic formulas using $\wedge, \vee, \neg$ and both number and string quantifiers $\exists x, \exists X, \forall x, \forall X$. Bounded number quantifiers are defined as usual, and the bounded string quantifier $\exists X \le t\ \varphi$ stands for $\exists X(|X| \le t \wedge \varphi)$ and $\forall X \le t\ \varphi$ stands for $\forall X(|X| \le t \supset \varphi)$, where X does not occur in the term t.

A structure for $\mathcal{L}^2_A$ is defined in the same way as a structure for a single-sorted language, except now there are two nonempty domains U_1 and U_2, one for numbers and one for strings. Each symbol of $\mathcal{L}^2_A$ is interpreted in $\langle U_1, U_2 \rangle$ by a relation or function of appropriate type, with $=^1$ and $=^2$ interpreted as true equality on U_1 and U_2, respectively. In the standard structure $\underline{\mathbb{N}}_2$, U_1 is $\mathbb{N}$ and U_2 is the set of finite subsets of $\mathbb{N}$. Each symbol of $\mathcal{L}^2_A$ gets its intended interpretation.

In general we will consider a vocabulary $\mathcal{L}$ which extends $\mathcal{L}^2_A$. We require that the bounding terms (for the bounded quatifiers) are restricted to mention the functions of $\mathcal{L}^2_A$ only. A formula is $\mathbf{\Sigma}^B_0(\mathcal{L})$ if it has no string quantifiers and all number quantifiers are bounded. A formula is $\mathbf{\Sigma}^B_1(\mathcal{L})$ ($\mathbf{\Pi}^B_1(\mathcal{L})$, $\mathbf{\Sigma}^1_1(\mathcal{L})$, resp.) if it is a $\mathbf{\Sigma}^B_0(\mathcal{L})$ formula preceded by a block of quantifiers of the form $\exists X \le t$ ($\forall X \le t$, $\exists X$, resp.). If the block contains a single quantifier, the formula is also called single-$\mathbf{\Sigma}^B_1(\mathcal{L})$ (single-$\mathbf{\Pi}^B_1(\mathcal{L})$, single-$\mathbf{\Sigma}^1_1(\mathcal{L})$, resp.). A formula is $\mathbf{g\Sigma}^B_1(\mathcal{L})$ (resp. $\mathbf{g\Pi}^B_1(\mathcal{L})$) if it is obtained from $\mathbf{\Sigma}^B_0(\mathcal{L})$ formulas using the connectives $\wedge$ and $\vee$, bounded number quantifiers and bounded existential (resp. universal) string quantifiers ("**g**" for "general"). A formula is $\exists\mathbf{g\Sigma}^B_1(\mathcal{L})$ if it is of the form $\exists\vec{X}\varphi$, where φ is $\mathbf{g\Sigma}^B_1(\mathcal{L})$. We will omit $\mathcal{L}$ if it is $\mathcal{L}^2_A$.

The $\mathbf{\Sigma}^B_1$ formulas correspond to (in first-order logic) strict $\mathbf{\Sigma}^b_1$ formulas (i.e., $\mathbf{\Sigma}^b_1$ formulas where no bounded quantifier is inside the scope of any sharply bounded quantifier), while $\mathbf{g\Sigma}^B_1$ formulas correspond to $\mathbf{\Sigma}^b_1$ formulas. Similar for $\mathbf{\Pi}^B_1$ and $\mathbf{g\Pi}^B_1$ formulas.

2.2 Two-Sorted Complexity Classes

We study two-sorted versions of standard complexity classes, where the two sorts are those in the standard model $\underline{\mathbb{N}}_2$ for $\mathcal{L}^2_A$: the natural numbers and finite subsets of the natural numbers. When a class is defined in terms of machines or circuits, we assume that each number input is represented in unary notation (i.e., n is represented as a string of n 1's), and each finite subset X is represented by its characteristic bit string.

There are two kinds of functions: *number functions* and *string functions*. A number function $f(\vec{x}, \vec{X})$ takes values in $\mathbb{N}$, and a string function $F(\vec{x}, \vec{X})$ take values in finite subsets of $\mathbb{N}$. A function $f(\vec{x}, \vec{X})$ or $F(\vec{x}, \vec{X})$ is *polynomially bounded* (or *p–bounded*) if there is a polynomial $\mathbf{p}(\vec{x}, \vec{y})$ such that $f(\vec{x}, \vec{X}) < \mathbf{p}(\vec{x}, |\vec{X}|)$, or $|F(\vec{x}, \vec{X})| < \mathbf{p}(\vec{x}, |\vec{X}|)$. The functions classes we consider here contain only p–bounded functions.

The class (uniform) $\mathbf{AC}^0$ can be characterized as the set of relations $R(\vec{x}, \vec{X})$ which are accepted by alternating Turing machines in time $O(\log(n))$ with constant alternations. The following result is from [17, 10].

Theorem 2.1. *A relation $R(\vec{x}, \vec{X})$ is in $\mathbf{AC}^0$ iff it is represented by some $\mathbf{\Sigma}^B_0$ formula $\varphi(\vec{x}, \vec{X})$.*

We define $\mathbf{AC}^0$ reducibility in the "Turing" style, as opposed to the many-one style. The idea is (see for example [2]) that F is $\mathbf{AC}^0$ reducible to $\mathcal{L}$ if F can be computed by a (uniform) polynomial size constant depth family of circuits which have unbounded fanin gates computing functions from $\mathcal{L}$, in addition to Boolean gates. We follow [11] and make this precise in Definition 2.3 below, based on Theorem 2.1.

The *bit graph* $B_F(z, \vec{x}, \vec{X})$ of a string function $F(\vec{x}, \vec{X})$ is defined by the condition

$$B_F(z, \vec{x}, \vec{X}) \equiv F(\vec{x}, \vec{X})(z) \tag{1}$$

Definition 2.2. *A string function is* $\mathbf{\Sigma}^B_0$–definable *from a collection $\mathcal{L}$ of two-sorted functions and relations if it is p–bounded and its bit graph is represented by a $\mathbf{\Sigma}^B_0(\mathcal{L})$ formula. Similarly, a number function is* $\mathbf{\Sigma}^B_0$–definable *from $\mathcal{L}$ if it is p–bounded and its graph is represented by a $\mathbf{\Sigma}^B_0(\mathcal{L})$ formula.*

Definition 2.3. *We say that a string function F (resp. a number function f) is* $\mathbf{AC}^0$ *reducible to* $\mathcal{L}$ *if there is a sequence of string functions* $F_1, \ldots, F_n$ $(n \geq 0)$ *such that*

$$F_i \text{ is } \mathbf{\Sigma}_0^B\text{–definable from } \mathcal{L} \cup \{F_1, \ldots, F_{i-1}\}, \text{ for } i = 1, \ldots, n; \tag{2}$$

and that F (resp. f) is $\mathbf{\Sigma}_0^B$*–definable from* $\mathcal{L} \cup \{F_1, \ldots, F_n\}$*. A relation R is* $\mathbf{AC}^0$ *reducible to* $\mathcal{L}$ *if there is a sequence* $F_1, \ldots, F_n$ *as above, and R is represented by a* $\mathbf{\Sigma}_0^B(\mathcal{L} \cup \{F_1, \ldots, F_n\})$ *formula.*

The uniform classes $\mathbf{TC}^0$, $\mathbf{AC}^0(m)$, and $\mathbf{ACC}$ can be defined in several equivalent ways [2]. Here we define them using $\mathbf{AC}^0$ reducibility, and later we characterize them using generalized quantifiers.

Definition 2.4 ($\mathbf{TC}^0$, $\mathbf{AC}^0(m)$, $\mathbf{ACC}$). *Let $numones(z, X)$ be the number of elements of X which are less than z.* [2] *Then* $\mathbf{TC}^0$ *is the class of relations* $\mathbf{AC}^0$ *reducible to numones. Similarly, for each $m \in \mathbb{N}$, $m \geq 2$, let $mod_m(z, X) = (numones(z, X) \mod m)$. Then* $\mathbf{AC}^0(m)$ *is the class of relations* $\mathbf{AC}^0$ *reducible to mod_m. The class* $\mathbf{ACC}$ *is the union of all* $\mathbf{AC}^0(m)$*, for $m \geq 2$.*

In general, each two-sorted relation class $\mathbf{C}$ is associated with a function class $\mathbf{FC}$. A number function belongs to $\mathbf{FC}$ if it is p–bounded, and its graph is in $\mathbf{C}$. A string function $F(\vec{x}, \vec{X})$ belongs to $\mathbf{FC}$ if it is p–bounded, and its bit graph (1) is in $\mathbf{C}$.

Lemma 2.5. $\mathbf{FTC}^0$ *is the class of functions* $\mathbf{AC}^0$ *reducible to numones. For $m \in \mathbb{N}, m \geq 2$,* $\mathbf{FAC}^0(m)$ *is the class of functions* $\mathbf{AC}^0$ *reducible to mod_m.* $\mathbf{FACC}$ *is the class of functions* $\mathbf{AC}^0$ *reducible to mod_m, for some m.*

Another characterization of $\mathbf{TC}^0$ is as follows. Consider augmenting the current two-sorted logic with the *counting* quantifier, i.e.,

$$\exists s\ z < t \varphi(z, \vec{x}, \vec{X})$$

is a formula which is true if and only if there are exactly s values of z such that $\varphi(z, \vec{x}, \vec{X})$ is true. It has been shown [25, Theorem 2.9] that $\mathbf{TC}^0$ is exactly the class of relations represented by $\mathbf{\Sigma}_0^{B,COUNT}$ formulas, i.e., bounded formulas which allow only number quantifiers and the counting quantifier. The proof in [25] is based on the characterization of $\mathbf{TC}^0$ [2]: $\mathbf{TC}^0 = \mathbf{FO}(COUNT)$. It shows how to translate $\mathbf{FO}(COUNT)$ formulas into $\mathbf{\Sigma}_0^{B,COUNT}$ formulas, and vice versa.

2.3 The Threshold Quantifier and Threshold Operation

Observe that the counting quantifier as discussed above "counts" *exactly* the number of z's that make $\varphi(z)$ true. We now define the *threshold* quantifier, which has syntax

$$\exists^{\geq} s\ z < t \varphi(z) \tag{3}$$

where s, t are terms not containing z. (The variable z is bound by the quantifier.) The semantics is given by the condition that (3) holds if and only if there are *at least* s values of z less than t that make φ true. This is similar to the counting quantifier, but we find the threshold quantifier more convenient, and will use it here.

Let $\mathbf{\Sigma}_0^{B,Th}$ be the class of formulas built in the same way as $\mathbf{\Sigma}_0^B$, except now we allow threshold quantifiers in addition to bounded number quantifiers. The following result and its corollary provide interesting characterizations of $\mathbf{TC}^0$, but they are not used in the rest of this paper.

Theorem 2.6. $\mathbf{TC}^0$ *is the class of relations represented by* $\mathbf{\Sigma}_0^{B,Th}$ *formulas.*

[2] Thus the number of elements of X is $numones(|X|, X)$.

Proof. First, let $\varphi(\vec{x},\vec{X})$ be a $\mathbf{\Sigma}_0^{B,Th}$ formula. We will prove by induction on the structure of φ that it represents a $\mathbf{TC}^0$ relation. The base case where φ is an atomic formula is straightforward. For the induction step, consider the interesting case where

$$\varphi(\vec{x},\vec{X}) \equiv \exists^{\geq} s\ z < t\ \varphi'(z,\vec{x},\vec{X}).$$

By the induction hypothesis, $\varphi'(z,\vec{x},\vec{X})$ represents a $\mathbf{TC}^0$ relation. In other words, it represents the same relation as some $\mathbf{\Sigma}_0^B(\{numones, F_1,\ldots,F_n\})$ formula $\psi(z,\vec{x},\vec{X})$, for a sequence $F_1,\ldots,F_n$ of string functions satisfying (2). Let F_{n+1} be $\mathbf{\Sigma}_0^B$–definable from $\{numones, F_1,\ldots,F_n\}$ as follows:

$$F_{n+1}(\vec{x},\vec{X})(z) \Leftrightarrow z < t \wedge \psi(z,\vec{x},\vec{X}).$$

Then φ represents the same relation as the formula

$$numones(t, F_{n+1}(\vec{x},\vec{X})) \geq s.$$

For the other direction, we will prove by induction on $n \geq 0$ a stronger result:

If $F_1,\ldots,F_n$ is any sequence of string functions satisfying (2) (with $\mathcal{L} = \{numones\}$), then for any $\mathbf{\Sigma}_0^{B,Th}(\{numones, F_1,\ldots,F_n\})$ formula $\psi(\vec{x},\vec{X})$ there is a $\mathbf{\Sigma}_0^{B,Th}$ formula $\varphi(\vec{x},\vec{X})$ that represents the same relation. (*)

a) For the base case, we prove by induction on the structure of a $\mathbf{\Sigma}_0^{B,Th}(numones)$ formula ψ that there is a $\mathbf{\Sigma}_0^{B,Th}$ formula φ that represents the same relation as ψ. It suffices to show that any atomic formula $\psi(numones)$ (i.e., ψ contains $numones$) is equivalent to a $\mathbf{\Sigma}_0^{B,Th}$ formula. Let $numones(t_1,X_1),\ldots,numones(t_m,X_m)$ be all occurrences of $numones$ in ψ, enumerated in some order such that if $numones(t_i,X_i)$ is a subterm of t_j then $i<j$. Let $u_1,\ldots,u_m$ be a list of new variables. Let t'_j be the result of replacing each maximal subterm $numones(t_i,X_i)$ of t_j by u_i and let ψ' be the result of replacing each maximal subterm $numones(t_i,X_i)$ in ψ by u_i. Note that if u_i occurs in t'_j then $i<j$. Now the $\mathbf{\Sigma}_0^{B,Th}$ formula φ is

$$\exists u_1 \leq t'_1 \ldots \exists u_m \leq t'_m,\ \psi' \wedge \bigwedge_{k=1}^{m} [\exists^{\geq} u_k\ z < t'_k\ X_k(z) \wedge \neg\exists^{\geq}(u_k+1)\ z < t'_k\ X_k(z)].$$

b) For the induction step, suppose that F_{n+1} is $\mathbf{\Sigma}_0^B$–definable from $\{numones, F_1,\ldots,F_n\}$ (for $n \geq 0$). Let ψ be a $\mathbf{\Sigma}_0^{B,Th}(\{numones, F_1,\ldots,F_{n+1}\})$ formula. We will show how to eliminate F_{n+1} from ψ by induction on the depth of nesting of F_{n+1} in ψ.

For a term or formula ω, we define $d(\omega)$ to be the maximum depth of nesting of any occurrence of F_{n+1} in ω.

We will prove the following by induction on $k \geq 0$:

If $d(\psi) = k$, then there is a $\mathbf{\Sigma}_0^{B,Th}$ formula φ that represents the same relation as ψ. (**)

(i) The base case where $k=0$ follows from the induction hypothesis of (*), since there is no occurrence of F_{n+1} in ψ.

(ii) Suppose that (**) holds for all ψ where $d(\psi) \leq k$. It suffices to prove (**) when ψ is an atomic formula, and $d(\psi) = k+1$.

Let $F_{n+1}(\vec{r}_1,\vec{T}_1),\ldots,F_{n+1}(\vec{r}_\ell,\vec{T}_\ell)$ be all string terms in ψ of the form $F_{n+1}(\vec{r},\vec{T})$, where $d(F_{n+1}(\vec{r},\vec{T})) = k+1$. (Thus $max(\{d(r) \mid r \in \vec{r}_i\} \cup \{d(T) \mid T \in \vec{T}_i\}) = k$, for $i = 1,\ldots,\ell$.) Let $W_1,\ldots,W_\ell$ be new string variables, and let ψ' be the formula obtained from ψ by replacing each $F_{n+1}(\vec{r}_i,\vec{T}_i)$ with W_i, $i=1,\ldots,\ell$. Then ψ' is atomic, since ψ is atomic. Since $d(\psi') \leq k$, it follows by the induction hypothesis that there is a $\mathbf{\Sigma}_0^{B,Th}$

formula θ that represents the same relation as ψ'. Then each W_i can occur in θ only in the form $|W_i|$, or $W_i(r)$, for some number term r (r might contain some $|W_j|$'s).

Suppose that F_{n+1} is defined by

$$F_{n+1}(\vec{x},\vec{X})(z) \;\leftrightarrow\; z < t_{n+1}(\vec{x},\vec{X}) \wedge \varphi_{n+1}(z,\vec{x},\vec{X})$$

where t_{n+1} is a term in the base language $\mathcal{L}_A^2$ and ϕ_{n+1} is a $\mathbf{\Sigma}_0^B(\{numones, F_1, ..., F_n\})$ formula. Now each occurrence of $|W_i|$ in θ can be eliminated by the equivalence

$$|F_{n+1}(\vec{r}_i,\vec{T}_i)| = z_i \;\leftrightarrow\; \delta_i$$

where $z_1, ..., z_\ell$ are new number variables and (setting $t'_i \equiv t_{n+1}(\vec{r}_i,\vec{T}_i)$)

$$\delta_i \;\equiv\; z_i \le t'_i \wedge [\forall x < t'_i,\ x \ge z_i \supset \neg\varphi_{n+1}(x,\vec{r}_i,\vec{T}_i)] \wedge [z_i > 0 \supset \varphi_{n+1}(z_i - 1,\vec{r}_i,\vec{T}_i)]$$

(Note that $d(\varphi_{n+1}(z,\vec{r}_i,\vec{T}_i)) \le k$.) Let θ' be obtained from θ by replacing each $|W_i|$ by z_i. Let θ'' be the formula

$$\exists z_1 \le t'_1 \ldots \exists z_\ell \le t'_\ell(\theta' \wedge \delta_1 \wedge ... \wedge \delta_\ell)$$

Next, replace each occurrence of the form $W_i(r)$ in θ'' (such r does not contain any of the W_j's) with

$$r < t'_i \wedge \varphi_{n+1}(r,\vec{r}_i,\vec{T}_i).$$

Let θ''' be the resulting formula. Then θ''' represents the same relation as ψ. Since $d(\theta''') \le k$, we can apply the induction hypothesis to θ''' to obtained the desired $\mathbf{\Sigma}_0^{B,Th}$ formula φ that represents the same relation as ψ. □

Define the *threshold* operation as follows. It takes a relation $Q(z)$ (which may contain other parameters) to the relation $[\mathsf{Th}zQ](k,y)$ defined by

$$[\mathsf{Th}zQ](k,y) \Leftrightarrow \text{ there are at least } k \text{ values of } z < y \text{ that satisfy } Q(z).$$

Then the threshold and Boolean operations together simulate the bounded number quantification operations. For example, the relation $[\exists z \le tQ](z)$ is the same as the relation

$$[\mathsf{Th}zQ](1,t).$$

The following is immediate from Theorem 2.6.

Corollary 2.7. *$\mathbf{TC}^0$ is the closure of $\mathbf{AC}^0$ relations under the threshold and Boolean operations.*

2.4 The Modulo m Quantifier and Operation

For each $m \in \mathbb{N}, m \ge 2$, the *modulo m quantifier* and *modulo m operation* can be defined similarly as the threshold quantifier and threshold operation, with a little more complication. In particular, the modulo m quantifier Mod_m only makes sense when the variable it quantifies over is bounded. Thus,

$$\mathsf{Mod}_m z < t\ \varphi(z)$$

is true if and only if the number of $z < t$ satisfying $\varphi(z)$ is exactly 1 modulo m. Similarly, the modulo m operation takes a relation $Q(z,\vec{x},\vec{X})$ into the relation $[\mathsf{Mod}_m zQ](y,\vec{x},\vec{X})$ which consists of all tuples $(y,\vec{x},\vec{X})$ such that

$$|\{(z,\vec{x},\vec{X}) : z < y \text{ and } Q(z,\vec{x},\vec{X})\}| = 1 \mod m.$$

Let $\mathbf{\Sigma}_0^{B,\mathsf{Mod}_m}$ formulas be bounded formulas in our two-sorted logic augmented with the Mod_m quantifier, where only bounded number quantifiers are allowed. Then the analog of Theorem 2.6 and Corollary 2.7 can be proved by slight modifications of the original proofs.

Theorem 2.8. *For each $m \in \mathbb{N}, m \geq 2$, $\mathbf{AC}^0(m)$ is the class of relations represented by $\mathbf{\Sigma}_0^{B,\mathsf{Mod}_m}$ formulas. It is also the closure of $\mathbf{AC}^0$ relations under Boolean, bounded number quantification and modulo m operations.*

3 The Theories

3.1 The Theory $\mathbf{V}^0$

We start by describing the theory $\mathbf{V}^0$ [10, 11] for the complexity class $\mathbf{AC}^0$. All of the theories that we introduce here are extensions of $\mathbf{V}^0$.

The theory $\mathbf{V}^0$ has underlying language $\mathcal{L}_A^2$ and is axiomatized by the set of axioms 2-**BASIC** and the $\mathbf{\Sigma}_0^B$-**COMP** axiom scheme. First, 2-**BASIC** is the set of the axioms **B1** – **B12**, **L1**, **L2** and **SE** below.

B1. $x + 1 \neq 0$
B2. $x + 1 = y + 1 \supset x = y$
B3. $x + 0 = x$
B4. $x + (y + 1) = (x + y) + 1$
B5. $x \cdot 0 = 0$
B6. $x \cdot (y + 1) = (x \cdot y) + x$
B7. $(x \leq y \wedge y \leq x) \supset x = y$
B8. $0 \leq x$
B9. $x \leq y \wedge y \leq z \supset x \leq z$
B10. $x \leq y \vee y \leq x$
B11. $x \leq y \leftrightarrow x < y + 1$
B12. $x \neq 0 \supset \exists y < x(y + 1 = x)$
L1. $X(y) \supset y < |X|$
L2. $y + 1 = |X| \supset X(y)$
SE. $X = Y \leftrightarrow [|X| = |Y| \wedge \forall z < |X|(X(z) \leftrightarrow Y(z))]$

The axiom scheme $\mathbf{\Sigma}_0^B$-**COMP** is the set of all formula of the form

$$\exists X \leq a \forall z < a,\ X(z) \leftrightarrow \varphi(z), \tag{4}$$

where φ is a $\mathbf{\Sigma}_0^B$ formula not containing X.

Although $\mathbf{V}^0$ does not have an explicit induction scheme, axioms **L1** and **L2** tell us that if X is nonempty then it has a largest element, and thus we can show that $\mathbf{V}^0$ proves the X-**MIN** formula

$$0 < |X| \supset \exists x < |X|(X(x) \wedge \forall y < x\ \neg X(y))$$

and X-**IND**

$$[X(0) \wedge \forall y < z(X(y) \supset X(y + 1))] \supset X(z)$$

From this and $\mathbf{\Sigma}_0^B$-**COMP** we conclude that $\mathbf{V}^0$ proves the scheme

$$\mathbf{\Sigma}_0^B\text{-}\mathbf{IND}: \qquad [\varphi(0) \wedge \forall x(\varphi(x) \supset \varphi(x + 1))] \supset \forall z \varphi(z)$$

where $\varphi(x)$ is any $\mathbf{\Sigma}_0^B$ formula (possibly containing parameters).

A pairing function can be defined in $\mathbf{V}^0$ by using $\langle x, y\rangle$ to abbreviate the term $(x + y)(x + y + 1) + 2y$. Then $\mathbf{V}^0$ proves that the map $(x, y) \mapsto \langle x, y\rangle$ is an one-one map from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. We use this idea to define a binary array X using the definition $X(x, y) \equiv X(\langle x, y\rangle)$. By iterating the pairing function we can define a multidimensional array $X(\vec{x})$. Then $\mathbf{V}^0$ proves the corresponding comprehension scheme

$$\exists Z \leq \langle \vec{a}\rangle \forall \vec{z} < \langle \vec{a}\rangle, Z(\vec{z}) \leftrightarrow \varphi(\vec{z})$$

for any $\mathbf{\Sigma}_0^B$ formula φ.

If we think of Z as a two-dimensional array, then we can represent row x in this array by $Z^{[x]}$, where $Z^{[x]} = \mathit{Row}(x, Z)$ is the $\mathbf{FAC}^0$ string function with bit-defining axiom

$$Z^{[x]}(i) \leftrightarrow \mathit{Row}(x, Z)(i) \leftrightarrow i < |Z| \wedge Z(x, i) \tag{5}$$

Lemma 3.1. *Let $\mathbf{V}^0(Row)$ be the extension of $\mathbf{V}^0$ obtained by adding the function Row with defining axiom (5). Then $\mathbf{V}^0(Row)$ is conservative over $\mathbf{V}^0$, and every $\mathbf{\Sigma}_0^B(Row)$ formula φ is provably equivalent in $\mathbf{V}^0(Row)$ to a $\mathbf{\Sigma}_0^B$ formula φ'.*

Proof. Conservativity follows from the fact that Row is $\mathbf{\Sigma}_1^B$-definable in $\mathbf{V}^0$(see Lemma 3.5).

For the second part, we may assume by the axiom **SE** that φ does not contain $=^2$. We proceed by induction on the maximum nesting depth of Row in φ. It suffices to consider the case in which φ is atomic. If φ has the form $Row(t,T)(s)$, then φ is equivalent to $T(t,s)$ which by the induction hypothesis is equivalent to a $\mathbf{\Sigma}_0^B$ formula.

Now suppose that φ is atomic and does not have the form $Row(t,T)(s)$. Let $Row(t_1,T_1),...,Row(t_k,T_k)$ be the maximal depth string terms occurring in φ. Then each such term $Row(t_i,T_i)$ must occur in the context $|Row(t_i,T_i)|$, so

$$\varphi \equiv \varphi'(|Row(t_1,T_1)|,...,|Row(t_k,T_k)|)$$

where $\varphi'(x_1,...,x_k)$ has less Row-nesting depth than φ. Then $\mathbf{V}^0(Row)$ proves

$$\varphi \leftrightarrow \exists x_1 \leq |T_1|...\exists x_k \leq |T_k|,(\bigwedge_{i=1}^{k} row\text{-}length(x_i,t_i,T_i)) \wedge \varphi(x_1,...,x_k)$$

where $row\text{-}length(x,y,Z)$ is a $\mathbf{\Sigma}_0^B$ formula expressing the condition $x = |Row(y,Z)|$. We can now apply the induction hypothesis to the RHS. ☐

3.2 The Theory $\mathbf{VTC}^0$

The theory $\mathbf{VTC}^0$ is $\mathbf{V}^0$ together with *NUMONES*, which is essentially a $\mathbf{\Sigma}_1^B$ defining axiom for *numones* (Definition 2.4) . Let $\varphi_{NUMONES}(X,Y)$ be the $\mathbf{\Sigma}_0^B$ formula stating that Y is a counting array of X, i.e., for each $z \leq |X|$, $Y(z,y)$ holds if and only if $numones(z,X) = y$:

$$\begin{aligned}\varphi_{NUMONES}(X,Y) \equiv {} & [\forall z \leq |X| \exists! y \leq |X| Y(z,y)] \ \wedge \ Y(0,0) \ \wedge \\ & \forall z < |X| \forall y \leq |X|,\ Y(z,y) \supset [(X(z) \supset Y(z+1,y+1)) \wedge (\neg X(z) \supset Y(z+1,y))].\end{aligned} \tag{6}$$

Definition 3.2. *Let NUMONES denote $\forall X \exists Y \varphi_{NUMONES}(X,Y)$. The theory $\mathbf{VTC}^0$ is $\mathbf{V}^0$ extended by the axiom NUMONES.*

Note that $\mathbf{V}^0$ proves that *NUMONES* implies that same axiom with $\exists Y$ replaced by the bounded quantifier $\exists Y \leq 1 + \langle |X|, |X| \rangle$. Hence $\mathbf{VTC}^0$ is equivalent to a theory with bounded axioms.

Lemma 3.3. *The theories $\mathbf{V}^0$ and $\mathbf{VTC}^0$ are finitely axiomatizable.*

Proof. The finite axiomatizability of $\mathbf{V}^0$ is proved in [12]. The theory $\mathbf{VTC}^0$ is the result of adding a single axiom to $\mathbf{V}^0$. ☐

The next definition refers to the notion of $\mathbf{\Sigma}_1^1(\mathcal{L})$ formula, defined in Section 2.1.

Definition 3.4. *Let $\mathcal{T}$ be an extension of $\mathbf{V}^0$ over a language $\mathcal{L}$. A string function $F(\vec{x},\vec{X})$ is $\Sigma_1^1(\mathcal{L})$-*definable *in $\mathcal{T}$ if it satisfies*

$$Y = F(\vec{x},\vec{X}) \leftrightarrow \varphi(\vec{x},\vec{X},Y) \tag{7}$$

for some $\Sigma_1^1(\mathcal{L})$ formula φ, and

$$\mathcal{T} \vdash \forall\vec{x}\forall\vec{X}\exists! Y \phi(\vec{x},\vec{X},Y) \tag{8}$$

The $\Sigma_1^1(\mathcal{L})$-definability for a number function $f(\vec{x},\vec{X})$ is defined similarly.

Lemma 3.5. *If $\mathcal{T}$ is an extension of $\mathbf{V}^0$ which satisfies (8) and F is not in the language of $\mathcal{T}$ and $\mathcal{T}'$ is the result of adding F to the language and adding (7) as an axiom, then F' is a conservative extension of $\mathcal{T}$.*

Proof. According to (8), every model of $\mathcal{T}$ has an expansion to a model of $\mathcal{T}'$ which satisfies (7). □

If $\mathcal{T}$ is a bounded theory, in the sense that the quantifiers in the axioms for $\mathcal{T}$ can be bounded by terms of $\mathcal{L}_A^2$, then by Parikh's Theorem [27, 10] it follows that a function is $\Sigma_1^1(\mathcal{L})$ definable in $\mathcal{T}$ iff it is Σ_1^B definable in $\mathcal{T}$.

We can now state one of our main results, which explains the sense in which our theories characterize the corresponding complexity classes. We already know [11] that the Σ_1^1-definable (and hence the Σ_1^B)-definable functions in $\mathbf{V}^0$ are precisely those in $\mathbf{FAC}^0$.

Theorem 3.6. *The Σ_1^1-definable (and the Σ_1^B-definable) functions in $\mathbf{VTC}^0$ are precisely those in $\mathbf{FTC}^0$.*

The proof is the subject of Subsections 3.3 - 3.5.

3.3 Universal Theories

We will employ the techniques from [11] to develop the universal version of our theories. Note that the axioms **B12** and **SE** are not universal statements. As in [11], **B12** is replaced by **B12′** and **B12″** below. Consider the number function pd where $pd(x)$ is the predecessor of x. Then **B12′** and **B12″** are the defining axioms of pd:

$$\mathbf{B12'}\ pd(0) = 0 \qquad \mathbf{B12''}\ x \neq 0 \supset pd(x) + 1 = x \tag{9}$$

The left–to–right direction of **SE** can be expressed by an open formula simply by replacing $\forall z < |X|$ by $z < |X| \supset$:

$$\mathbf{SE}' : X = Y \supset [|X| = |Y| \wedge z < |X| \supset (X(z) \leftrightarrow Y(z))].$$

The right–to–left direction of **SE** has an implicit quantifier $\exists z < |X|$. We can get rid of this by using the function $f_{\mathbf{SE}}$ (which is $f_{\alpha,t}$ in Definition 3.7 below, when $\alpha \equiv X(z) \leftrightarrow Y(z)$, and $t = |X|$):

$$f_{\mathbf{SE}}(X,Y) \leq |X|, \tag{11}$$
$$z < |X| \wedge (X(z) \not\leftrightarrow Y(z)) \supset X(f_{\mathbf{SE}}(X,Y)) \not\leftrightarrow Y(f_{\mathbf{SE}}(X,Y)), \tag{12}$$
$$z < f_{\mathbf{SE}}(X,Y) \supset X(z) \leftrightarrow Y(z). \tag{13}$$

Thus $f_{\mathbf{SE}}(X,Y)$ is the smallest number $< |X|$ which distinguishes X and Y, and $|X|$ if no such number exists. Let $\mathbf{SE}''$ be

$$\mathbf{SE}'' : (|X| = |Y| \wedge f_{\mathbf{SE}}(X,Y) = |X|) \supset X = Y.$$

Definition 3.7 ($\mathcal{L}_{\mathbf{FAC}^0}$)**.** *$\mathcal{L}_{\mathbf{FAC}^0}$ is the smallest class that satisfies*
a) *$\mathcal{L}_{\mathbf{FAC}^0}$ includes $\mathcal{L}_A^2 \cup \{pd, f_{\mathbf{SE}}\}$.*
b) *For each open formula $\alpha(z,\vec{x},\vec{X})$ over $\mathcal{L}_{\mathbf{FAC}^0}$ and term $t = t(\vec{x},\vec{X})$ of $\mathcal{L}_A^2$ there is a string function $F_{\alpha,t}$ of $\mathcal{L}_{\mathbf{FAC}^0}$ with defining axiom*

$$F_{\alpha,t}(\vec{x},\vec{X})(z) \leftrightarrow z < t \wedge \alpha(z,\vec{x},\vec{X}) \tag{15}$$

c) *For each open formula $\alpha(z,\vec{x},\vec{X})$ over $\mathcal{L}_{\mathbf{FAC}^0}$ and term $t = t(\vec{x},\vec{X})$ of $\mathcal{L}_A^2$ there is a number function $f_{\alpha,t}$ with defining axioms*

$$f_{\alpha,t}(\vec{x},\vec{X}) \leq t(\vec{x},\vec{X}) \tag{16}$$
$$f_{\alpha,t}(\vec{x},\vec{X}) < t(\vec{x},\vec{X}) \supset \alpha(f_{\alpha,t}(\vec{x},\vec{X}),\vec{x},\vec{X})) \tag{17}$$
$$z < f_{\alpha,t}(\vec{x},\vec{X}) \supset \neg\alpha(z,\vec{x},\vec{X}) \tag{18}$$

Note that $f_{\alpha,t}(\vec{x},\vec{X}) = \min z < t\, \alpha(z,\vec{x},\vec{X})$ and

$$\exists z < t\, \alpha(z,\vec{x},\vec{X}) \leftrightarrow f_{\alpha,t}(\vec{x},\vec{X}) < t. \tag{19}$$

We define the theory $\overline{\mathbf{V}}^0$ [11] to be the universal theory over $\mathcal{L}_{\mathbf{FAC}^0}$ whose axioms are the universal closures of the following list of open formulas: **B1 - B11**, **B12**$'$, **B12**$''$, **L1**, **L2**, **SE**$'$, **SE**$''$, the defining axioms (11), (12), (13) for $f_{\mathbf{SE}}$, and the defining axiom (15) for each function $F_{\alpha,t}$ and defining axioms (16), (17), (18) for each function $f_{\alpha,t}$.

Lemma 3.8. *For every $\mathbf{\Sigma}_0^B$ formula φ there is an open formula α of $\mathcal{L}_{\mathbf{FAC}^0}$ such that $\overline{\mathbf{V}}^0$ proves $(\varphi \leftrightarrow \alpha)$. For every open formula α of $\mathcal{L}_{\mathbf{FAC}^0}$ there is a $\mathbf{\Sigma}_0^B$ formula φ such that $\overline{\mathbf{V}}^0$ proves $(\varphi \leftrightarrow \alpha)$.*

Proof. The first sentence follows by structural induction on $\mathbf{\Sigma}_0^B$ formulas φ, using (19). To prove the second sentence consider an enumeration of the new function symbols of $\mathcal{L}_{\mathbf{FAC}^0}$ in some order such that the defining axioms of each function in the list mention only earlier functions in the list. Now show by induction on k that if α only involves functions occurring in the first k positions on the list then α is equivalent to some $\mathbf{\Sigma}_0^B$ formula φ. □

Theorem 3.9. *$\overline{\mathbf{V}}^0$ is a conservative extension of $\mathbf{V}^0$. The function symbols in $\mathcal{L}_{\mathbf{FAC}^0}$ represent precisely the functions in $\mathbf{FAC}^0$.*

Proof. To show that $\overline{\mathbf{V}}^0$ extends $\mathbf{V}^0$ it suffices to show $\overline{\mathbf{V}}^0$ proves the $\mathbf{\Sigma}_0^B$-**COMP** axioms (4). From the first sentence of Lemma 3.8 and (15) we have that for every $\mathbf{\Sigma}_0^B$ formula $\varphi(z,\vec{x},\vec{X})$ there is a function F in $\mathcal{L}_{\mathbf{FAC}^0}$ such that

$$\overline{\mathbf{V}}^0 \vdash \quad F(\vec{x},\vec{X})(z) \leftrightarrow z < a \wedge \varphi(z,\vec{x},\vec{X})$$

from which (4) follows.

To see that the extension is conservative we use the second sentence of Lemma 3.8 to show that every function of $\mathcal{L}_{\mathbf{FAC}^0}$ is $\mathbf{\Sigma}_1^B$-definable in $\mathbf{V}^0$. Finally it is clear from Definition 3.7 that the function symbols in $\mathcal{L}_{\mathbf{FAC}^0}$ represent precisely the functions in $\mathbf{FAC}^0$. □

It is worth emphasizing that $\overline{\mathbf{V}}^0$ proves the $\mathbf{\Sigma}_0^B$-**IND** and $\mathbf{\Sigma}_0^B$-**MIN** schemes, since it extends $\mathbf{V}^0$. This is true even though $\overline{\mathbf{V}}^0$ has purely universal axioms, and has no explicit induction axiom or rule.

Below we prove the General Witnessing Theorem for universal theories (Theorem 3.20). The Witnessing Theorems for our theories will follow from those of their corresponding universal conservative extensions. It follows that the $\exists g \mathbf{\Sigma}_1^B$-definable functions in these theories are in the appropriate complexity classes. For the other direction, it is clear that the universal theories define all functions in the appropriate classes, and Theorem 3.15 below shows the same for the original theories.

3.4 The Theory $\overline{\mathbf{VTC}}^0$

The function *numones* from Definition 2.4 has defining axioms

$$numones(0,X) = 0 \tag{20}$$

$$X(z) \supset numones(z+1,X) = numones(z,X)+1 \tag{21}$$

$$\neg X(z) \supset numones(z+1,X) = numones(z,X). \tag{22}$$

Since

$$y = numones(x,X) \leftrightarrow \exists Y,\ \varphi_{NUMONES}(X,Y) \wedge Y(x,y) \tag{23}$$

it is easy to see that *numones* is $\mathbf{\Sigma}_1^B$-definable in $\mathbf{VTC}^0$.

The vocabulary $\mathcal{L}_{\mathbf{FTC}^0}$ includes *numones* and is intended to represent the functions in $\mathbf{FTC}^0$.

Definition 3.10. *$\mathcal{L}_{\mathbf{FTC}^0}$ is defined in the same way as $\mathcal{L}_{\mathbf{FAC}^0}$ (Definition 3.7) with (a), (b) and (c) replaced by*
(a$'$) $\mathcal{L}_{\mathbf{FTC}^0}$ includes $\mathcal{L}_A^2 \cup \{pd, f_{\mathbf{SE}}, numones\}$,
(b$'$), (c$'$) are the same as (b), (c), except that $\mathcal{L}_{\mathbf{FAC}^0}$ is replaced by $\mathcal{L}_{\mathbf{FTC}^0}$.

The next lemma follows directly from the definitions.

Lemma 3.11. *The functions in $\mathcal{L}_{\mathbf{FTC}^0}$ represent precisely $\mathbf{FTC}^0$. A relation is in $\mathbf{TC}^0$ if and only if it is represented by some open $\mathcal{L}_{\mathbf{FTC}^0}$ formula.*

Definition 3.12. *$\overline{\mathbf{VTC}}^0$ is the universal theory over $\mathcal{L}_{\mathbf{FTC}^0}$ whose axioms are the universal closures of the following list of open formulas:* **B1 - B11**, **B12$'$**, **B12$''$**, **L1**, **L2**, **SE$'$**, **SE$''$**, *the defining axioms* (11), (12), (13) *for $f_{\mathbf{SE}}$, the defining axioms* (20), (21), (22) *for numones, and the defining axioms* (15) *and* (16), (17), (18) *for the functions of $\mathcal{L}_{\mathbf{FTC}^0}$.*

The first part of the analog of Lemma 3.8 is easily shown to hold in this context.

Lemma 3.13. *For every $\mathbf{\Sigma}_0^B(numones)$ formula φ there is an open formula α of $\mathbf{FTC}^0$ such that $\overline{\mathbf{VTC}}^0$ proves $(\varphi \leftrightarrow \alpha)$.*

From this we can show the following.

Lemma 3.14. *$\overline{\mathbf{VTC}}^0$ extends $\mathbf{VTC}^0$.*

Proof. Since $\overline{\mathbf{VTC}}^0$ extends $\overline{\mathbf{V}}^0$, it suffices to show $\overline{\mathbf{VTC}}^0$ proves *NUMONES*. We show this by pointing out that $\mathcal{L}_{\mathbf{FTC}^0}$ includes a string function F_{NUM} such that $\overline{\mathbf{VTC}}^0$ proves

$$Y = F_{NUM}(X) \supset \varphi_{NUMONES}(X, Y).$$

We can define F_{NUM} by the condition

$$F_{NUM}(X)(z, y) \leftrightarrow z \le |X| \wedge y = numones(z, X).$$

We can turn this into a proper bit-graph definition (15) by using a $\mathbf{\Sigma}_0^B(numones)$ formula and appealing to Lemma 3.13. □

Unfortunately the analog of the second sentence of Lemma 3.8 does not appear to hold. In general an open formula of $\mathcal{L}_{\mathbf{FTC}^0}$ is not equivalent to a $\mathbf{\Sigma}_0^B(numones)$ formula for the same reason that a $\mathbf{TC}^0$ circuit involving nested threshold gates cannot be made polynomially equivalent to a circuit with unnested threshold gates. Hence we must work harder to prove that $\overline{\mathbf{VTC}}^0$ is conservative over $\mathbf{VTC}^0$.

To prove conservativity, we note that $\overline{\mathbf{VTC}}^0$ can be obtained from $\mathbf{VTC}^0(numones)$ by successively adding $\mathbf{\Sigma}_0^B$-definable functions and their definitions. This fact together with Lemma 3.16 and the following theorem are used to show both that $\overline{\mathbf{VTC}}^0$ is conservative over $\mathbf{VTC}^0$ and that all functions in $\mathbf{FTC}^0$ are $\mathbf{\Sigma}_1^1$-definable in $\mathbf{VTC}^0$(Corollary 3.18).

Theorem 3.15. *Let $\mathcal{T}$ be an extension of $\mathbf{V}^0$ with a vocabulary $\mathcal{L}$ which includes the function Row, and suppose that $\mathcal{T}$ proves the defining equation (5) for Row. Suppose that $\mathcal{T}$ satisfies*
a) *$\mathcal{T}$ proves the $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** scheme, and*
b) *For each $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula α there is a $\mathbf{\Sigma}_1^B$ formula β such that $\mathcal{T} \vdash \alpha \leftrightarrow \beta$.*

Let $\mathcal{L}'$ extend $\mathcal{L}$ by adding a function symbol that is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$ (see Definition 2.2). Let $\mathcal{T}'$ be obtained from $\mathcal{T}$ by adding the function symbol and its defining axiom. Then $\mathcal{T}'$ is conservative over $\mathcal{T}$, and a) and b) hold with $\mathcal{T}$ replaced by $\mathcal{T}'$ and $\mathcal{L}$ replaced by $\mathcal{L}'$.

Proof. We will consider the case $\mathcal{L}' = \mathcal{L} \cup \{F\}$, where F is a string function $\mathbf{\Sigma}_0^B$ definable from $\mathcal{L}$, i.e., it has the defining axiom

$$F(\vec{x}, \vec{X})(u) \leftrightarrow u < t(\vec{x}, \vec{X}) \wedge \varphi(u, \vec{x}, \vec{X}) \tag{24}$$

for some $\mathcal{L}_A^2$ term t and $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula φ. The case in which $\mathcal{L}'$ extends $\mathcal{L}$ by a number function is handled similarly, except that number variables w_i are used instead of the string variables W_i in the argument below.

Since $\mathcal{T}$ proves the $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** scheme, it follows that it $\mathbf{\Sigma}_1^B(\mathcal{L})$–defines F. Therefore $\mathcal{T}'$ is conservative over $\mathcal{T}$.

a) We will show that $\mathcal{T}'$ proves a slightly modified version of the comprehension axiom

$$\exists Z \leq \langle \vec{b} \rangle \forall \vec{z} < b,\ Z(\vec{z}) \leftrightarrow \psi(\vec{z}) \tag{25}$$

for each $\mathbf{\Sigma}_0^B(\mathcal{L}')$ formula ψ, where $\vec{z}$ are all number free variables of ψ. It is straightforward to obtain the usual comprehension axiom scheme from this. Also, since $\mathcal{T}$ extends $\mathbf{V}^0$, it proves this version of $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP**. We will prove (25) by induction on the quantifier depth of ψ.

For the base case, ψ is quantifier-free. Suppose that $F(\vec{s}_1, \vec{T}_1), \ldots, F(\vec{s}_k, \vec{T}_k)$ are all occurrences of F in ψ. Note that the terms $\vec{s}_i, \vec{T}_i$ may contain $\vec{z}$ as well as F. Assume further that $\vec{s}_1, \vec{T}_1$ do not contain F, and for $1 < i \leq k$, any occurrence of F in $\vec{s}_i, \vec{T}_i$ must be of the form $F(\vec{s}_j, \vec{T}_j)$, for some $j < i$. We proceed to eliminate F from ψ by using its defining axiom (24).

Let $W_1, ..., W_k$ be new string variables. Let $\varphi_1(\vec{z}, u) \equiv \varphi(u, \vec{s}_1, \vec{T}_1)$, and for $2 \leq i \leq k$, $\varphi_i(\vec{z}, u)$ is obtained from $\varphi(u, \vec{s}_i, \vec{T}_i)$ by replacing every maximal occurrence of any $F(\vec{s}_j, \vec{T}_j)$, for $j < i$, by $W_j^{[\vec{z}]}$. Let t_i be obtained from $t(\vec{s}_i, \vec{T}_i)$ by the same procedure (for $i \leq k$). Thus F does not occur in any φ_i or t_i. Since $\mathcal{T}$ proves $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP**, it proves the existence of W_i such that

$$\forall \vec{z} < b,\ W_i^{[\vec{z}]}(u) \leftrightarrow u < t_i \wedge \varphi_i(\vec{z}, u) \qquad \text{for } 1 \leq i \leq k. \tag{26}$$

Let $\psi'(\vec{z}, W_1, \ldots, W_k)$ be obtained from $\psi(\vec{z})$ by replacing each maximal occurrence of $F(\vec{s}_i, \vec{T}_i)$ by $W_i^{[\vec{z}]}$, for $1 \leq i \leq k$. Then, by $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** and the fact that $\mathcal{L}$ contains *Row*,

$$\mathcal{T} \vdash \exists Z \leq \langle \vec{b} \rangle \forall \vec{z} < b,\ Z(\vec{z}) \leftrightarrow \psi'(\vec{z}, W_1, \ldots, W_k).$$

Then such Z satisfies $\forall \vec{z} < b,\ Z(\vec{z}) \leftrightarrow \psi(\vec{z})$ when each W_i is defined by (26).

For the induction step, it suffices to consider the case $\psi(\vec{z}) \equiv \forall x < t \varphi'(\vec{z}, x)$. By the induction hypothesis,

$$\mathcal{T}' \vdash \exists Z' \leq \langle \vec{b}, t \rangle \forall \vec{z} < b \forall x < t,\ Z'(\vec{z}, x) \leftrightarrow \varphi'(\vec{z}, x).$$

Now, by $\mathbf{\Sigma}_0^B$-**COMP**,

$$\mathbf{V}^0 \vdash \exists Z \leq \langle \vec{b} \rangle \forall \vec{z} < b,\ Z(\vec{z}) \leftrightarrow \forall x < t Z'(\vec{z}, x).$$

b) Suppose that

$$\alpha \equiv \mathrm{Q}_1 z_1 < r_1 \ldots \mathrm{Q}_n z_n < r_n \psi(\vec{z})$$

is a $\mathbf{\Sigma}_0^B(\mathcal{L}')$ formula, where ψ is quantifier-free. Let $\psi'(\vec{z}, W_1, \ldots, W_k)$ be obtained from $\psi(\vec{z})$ as described above in the proof of a). Define

$$\alpha'(W_1, ..., W_k) \equiv \mathrm{Q}_1 z_1 < r_1 \ldots \mathrm{Q}_n z_n < r_n \psi'(\vec{z}, W_1, ..., W_k).$$

For $1 \leq i \leq k$ let γ_i be the formula (26). Then, α is equivalent in $\mathcal{T}'$ to

$$\exists W_1 \leq \langle \vec{r}, t_1 \rangle \ldots \exists W_k \leq \langle \vec{r}, t_k \rangle,\ (\bigwedge \gamma_i) \wedge \alpha'(W_1, \ldots, W_k).$$

By property b) for $\mathcal{T}$ we may replace the part of the above formula following the string quantifier prefix by a $\mathbf{\Sigma}_1^B$ formula, and thus we obtain the required $\mathbf{\Sigma}_1^B$ formula β in c) for $\mathcal{T}'$. □

Let $\mathbf{VTC}^0(Row, numones)$ be $\mathbf{VTC}^0$ together with the functions *Row* and *numones* and their defining axioms (5), (20), (21), (22). Since both *Row* and *numones* are $\mathbf{\Sigma}_1^1$-definable in $\mathbf{VTC}^0$ it follows that $\mathbf{VTC}^0(Row, numones)$ is conservative over $\mathbf{VTC}^0$.

Lemma 3.16. *Let $\mathcal{T}$ be the theory $\mathbf{VTC}^0(Row, numones)$. Then $\mathcal{T}$ satisfies hypotheses a) and b) in Theorem 3.15.*

Proof. First note that $\mathbf{VTC}^0(Row)$ satisfies a) and b) by Lemma 3.1. We will prove the present lemma by modifying the proof of Theorem 3.15 applied as if $\mathcal{T}'$ is $\mathbf{VTC}^0(Row, numones)$ and $\mathcal{T}$ is $\mathbf{VTC}^0(Row)$.

Proceeding as in the proof of a), we want to show that $\mathcal{T}'$ proves (25) where $\psi(\vec{z})$ is a $\mathbf{\Sigma}_0^B(Row, numones)$ formula. Arguing as before, it suffices to consider the base case of the induction, where ψ is quantifier-free, and *numones* plays the role of F in the previous argument. Thus $numones(s_1, T_1), ..., numones(s_k, T_k)$ are all occurrences of *numones* in ψ, ordered as before. We proceed to eliminate the occurrences of *numones* from ψ using (23).

Let $w_1, ..., w_k$ be new number variables. Let $s'_1 \equiv s_1$, and for $2 \leq i \leq k$ let s'_i be obtained from s_i by replacing every maximal occurrence of $numones(s_j, T_j)$, for $j < i$, by w_j. Let T'_i be obtained from T_i in the same way. (Thus *numones* does not occur in any of the s_i's and T_i's.) Let $Y_1, ..., Y_k$ be new string variables. By Claim 3.17 (below), for $1 \leq i \leq k$, $\mathbf{VTC}^0(Row)$ proves the existence of Y_i such that

$$\forall \vec{z} < b \, \forall w_1 \leq s'_1 ... \forall w_k \leq s'_k \; \varphi_{NUMONES}(T'_i, Y_i^{[\vec{z}, \vec{w}]}). \tag{27}$$

If $Y_1, ..., Y_k$ each satisfies (27), and $w_1, ..., w_k$ each satisfies $Y_i^{[\vec{z},\vec{w}]}(s'_i, w_i)$ then by (23) each w_i must have its intended value $numones(s_i, T_i)$. Thus $\mathbf{VTC}^0(Row, numones)$ proves

$$\forall \vec{z} < b \forall \vec{w} \leq \vec{s'}, \; (\bigwedge_{i=1}^{k} (\varphi_{NUMONES}(T'_i, Y_i^{[\vec{z},\vec{w}]}) \wedge Y_i^{[\vec{z},\vec{w}]}(s'_i, w_i)) \;\supset\; \bigwedge_{i=1}^{k} w_i = numones(s_i, T_i)$$

Let $\psi'(\vec{z}, w_1, ..., w_k)$ be obtained from $\psi(\vec{z})$ by replacing each maximal occurrence of $numones(s_i, T_i)$ by w_i, for $1 \leq i \leq k$. Then by $\mathbf{\Sigma}_0^B(Row)$-$\mathbf{COMP}$ we have

$$\mathbf{V}^0(Row) \vdash \exists Z \leq \langle \vec{b} \rangle \forall \vec{z} < b, \; Z(\vec{z}) \leftrightarrow \exists w_1 \leq s'_1 ... \exists w_k \leq s'_k, \; (\bigwedge Y_i^{[\vec{z},\vec{w}]}(s'_i, w_i)) \wedge \psi'(\vec{z}, w_1, \ldots, w_k).$$

Then such Z satisfies $\forall \vec{z} < b, \; Z(\vec{z}) \leftrightarrow \psi(\vec{z})$ when each Y_i satisfies (27).

To prove b), suppose that

$$\alpha \equiv \mathrm{Q}_1 z_1 < r_1 \ldots \mathrm{Q}_n z_n < r_n \psi(\vec{z})$$

is a $\mathbf{\Sigma}_0^B(Row, numones)$ formula, where ψ is quantifier-free. Then, using the notation of the proof of a) above, $\mathbf{VTC}^0(Row, numones)$ proves

$$\alpha \leftrightarrow \exists \vec{Y} \leq \vec{t}, \; (\forall \vec{z} < b \forall \vec{w} \leq \vec{s'} \bigwedge \varphi_{NUMONES}(T'_i, Y_i^{[\vec{z},\vec{w}]})) \;\wedge\; \vec{Q}\vec{z} < \vec{r} \exists \vec{w} \leq \vec{s'}, \; \psi'(\vec{z}, \vec{w}) \wedge \bigwedge Y_i^{[\vec{z},\vec{w}]}(s'_i, w_i)$$

for suitable terms $\vec{t}$ bounding $\vec{Y}$. The RHS is a $\mathbf{\Sigma}_1^B(Row)$ formula which, by Lemma 3.1 is equivalent to a $\mathbf{\Sigma}_1^B$ formula. □

To complete the proof of Lemma 3.16, we show that for $1 \leq i \leq k$, $\mathbf{VTC}^0$ proves the existence of Y_i which satisfies (27). It suffices to show that $\mathbf{VTC}^0(Row)$ proves the existence of multiple "counting arrays" for polynomially many strings.

Claim 3.17. *The theory $\mathbf{VTC}^0(Row)$ proves the existence of Y such that $\forall u < b \; \varphi_{NUMONES}(X^{[u]}, Y^{[u]})$.*

Proof. We construct (using $\mathbf{\Sigma}_0^B(Row)$-**COMP**) multiple counting arrays $Y^{[0]}, \ldots, Y^{[b-1]}$ from the counting array Y' for a "big" string X', which is obtained from the strings $X^{[0]}, \ldots, X^{[b-1]}$ simply by concatenating them. More precisely, let X' be defined by

$$X'(u|X| + x) \leftrightarrow X^{[u]}(x), \quad \text{for } x < |X|, u < b.$$

Thus $X'(u|X|), \ldots, X'((u+1)|X| - 1)$ is a copy of $X^{[u]}$. Therefore

$$\mathit{numones}(z, X^{[u]}) = \mathit{numones}(u|X| + z, X') - \mathit{numones}(u|X|, X').$$

Let Y' be the counting array for X', i.e., $Y'(z,y) \Leftrightarrow \mathit{numones}(z, X') = y$. Then $Y^{[u]}(z,y) \Leftrightarrow y = \mathit{numones}(u|X| + z, X') - \mathit{numones}(u|X|, X')$. Hence

$$Y^{[u]}(z,y) \leftrightarrow \exists y_1, y_2 \leq |X'|,\ Y'(u|X|, y_1) \wedge Y'(u|X| + z, y_2) \wedge y + y_1 = y_2$$

$\square$

Corollary 3.18. *$\overline{\mathbf{VTC}}^0$ is a conservative extension of $\mathbf{VTC}^0$. Every function in $\mathcal{L}_{\mathbf{FTC}^0}$ is $\mathbf{\Sigma}_1^1$-definable in $\mathbf{VTC}^0$.*

Proof. $\mathbf{VTC}^0(Row, numones)$ is conservative over $\mathbf{VTC}^0$ because *Row* and *numones* are $\mathbf{\Sigma}_1^B$-definable in $\mathbf{VTC}^0$. According to Lemma 3.16 and Theorem 3.15, $\overline{\mathbf{VTC}}^0$ is the union of a sequence of conservative extension of $\mathbf{VTC}^0(Row, numones)$ satisfying a) and b). Thus (by compactness) $\overline{\mathbf{VTC}}^0$ is conservative over $\mathbf{VTC}^0$. Each of these extensions is obtained by adding a $\mathbf{\Sigma}_0^B(\mathcal{L})$-definable function. The graph of each such function has a $\mathbf{\Sigma}_0^B(\mathcal{L})$ definition, which by b) is provably equivalent to a $\mathbf{\Sigma}_1^B$ formula (in the language of $\mathbf{VTC}^0$). Hence this function is $\mathbf{\Sigma}_1^B$-definable in $\overline{\mathbf{VTC}}^0$ and hence in $\mathbf{VTC}^0$. $\square$

The above corollary proves one direction of Theorem 3.6 for the case of $\mathbf{VTC}^0$. For the other direction we need witnessing theorems, which are the subject of Subsection 3.5.

Recall that each string function $F \in \mathbf{FTC}^0$ has a defining axiom according to our construction of $\mathcal{L}_{\mathbf{FTC}^0}$ (see Definition 3.10 and Lemma 3.11). In fact, there is a finite sequence of $\mathbf{FTC}^0$ functions $F_1, \ldots, F_n$ that are involved in defining F. Let $\mathcal{L}(F)$ denote this sequence of functions (including F), and let $\mathrm{AX}(\mathcal{L}(F))$ be the set of their defining axioms. The following corollary is proved similarly to Corollary 3.18.

Corollary 3.19. *For each $F \in \mathbf{FTC}^0$, $\overline{\mathbf{VTC}}^0$ is a conservative extension of the theory $\mathbf{VTC}^0 \cup \mathrm{AX}(\mathcal{L}(F))$.*

3.5 Witnessing Theorems

In this subsection we will prove the remaining direction of Theorem 3.6, namely that the $\mathbf{\Sigma}_1^1$-definable functions in each of our various theories are in the appropriate complexity class.

We will use the proof system $\mathbf{LK}^2$ (see e.g., [10]) which extends $\mathbf{LK}$ (see also [7]) by the introduction rules for string variables quantifiers. It is convenient to distinguish between *bound variables* (which are denoted by $x, y, z, \ldots$ for number variables, and $X, Y, Z, \ldots$ for the string variables) and *free variables* (which are denoted by $a, b, c, \ldots$ for number variables, and $\alpha, \beta, \gamma, \ldots$ for the string variables). Recall the definition of $\mathcal{L}_{\mathbf{FAC}^0}$ in Definition 3.7.

Theorem 3.20 (General Witnessing Theorem). *Suppose that $\mathcal{L}$ extends $\mathcal{L}_{\mathbf{FAC}^0}$ and that it satisfies conditions* b, c *in Definition 3.7 with $\mathcal{L}_{\mathbf{FAC}^0}$ replaced by $\mathcal{L}$. Suppose that $\mathcal{T}$ is an open theory extending $\overline{\mathbf{V}}^0$ and that $\mathcal{T}$ contains the defining axiom* (15) *for each function $F_{\alpha,t}$ of $\mathcal{L}$, and the defining axioms* (16), (17) *and* (18) *for each function $f_{\alpha,t}$ of $\mathcal{L}$. Then for each theorem $\exists \vec{Z} \varphi(\vec{a}, \vec{\alpha}, \vec{Z})$ of $\mathcal{T}$, where φ is a $\mathbf{g\Sigma}_1^B$ formula, there are functions $\vec{F}$ of $\mathcal{L}$ such that*

$$\mathcal{T} \vdash \forall \vec{x} \forall \vec{X}\ \varphi(\vec{x}, \vec{X}, \vec{F}(\vec{x}, \vec{X})).$$

Proof. Note that when φ is an open formula, the Theorem is an application of Herbrand Theorem. To prove the current Theorem for the general case we will follow the proof theoretic approach and examine the $\mathbf{LK}^2$–$\mathcal{T}$ proofs (i.e., proofs in $\mathbf{LK}^2$ with nonlogical axioms from $\mathcal{T}$). In particular, we will explicitly witness the string existential quantifiers in every line of an *anchored* proofs of $\exists\vec{Z}\varphi(\vec{a},\vec{\alpha},\vec{Z})$ by functions from $\mathcal{L}$. This extends the proof of the Witnessing Theorem for $\mathbf{V}^0$ presented in [10]. First, the following Claim will simplify our arguments.

Claim: For each $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula $\phi(\vec{x},\vec{X})$, there is an open formula $\psi(\vec{x},\vec{X})$ of $\mathcal{L}$ such that

$$\mathcal{T} \vdash \psi(\vec{x},\vec{X}) \leftrightarrow \phi(\vec{x},\vec{X}).$$

Note that on page 10 we have used $f_{\mathbf{SE}}$ to eliminate an implicit quantifier $\exists z < |X|$ in the axiom **SE**. The proof of this Claim is similar and is omitted.

Now for simplicity, assume that we are to witness a single variable Z in $\exists Z\varphi(\vec{x},\vec{X},Z)$. Consider the most interesting case when φ is of the form:

$$\varphi(\vec{a},\vec{\alpha},Z) \equiv \forall y_n < b\ \exists Y_n < b\ \ldots\ \forall y_1 < b\ \exists Y_1 < b\ \theta(\vec{a},\vec{\alpha},\vec{y},\vec{Y},Z)$$

By the above Claim we can assume that θ is an open formula of $\mathcal{L}$.

An *anchored* $\mathbf{LK}^2$–$\mathcal{T}$ proof π is a proof in the system $\mathbf{LK}^2$ with additional nonlogical axioms the instances of axioms of $\mathcal{T}$, and the cut formulas of π are restricted to these instances only.[3] By the *Anchored Proof Theorem* [10] there exists an anchored $\mathbf{LK}^2$–$\mathcal{T}$ proof π of $\exists Z\ \varphi(\vec{a},\vec{\alpha},Z)$. Thus quantified formulas in π can appear only in the succedents, and must be of one of the two forms below (we will not mention the bound b on variables)

$$\exists Y_m\ \forall y_{m-1}\ldots\ \forall y_1\ \exists Y_1\ \theta(\vec{a},\vec{\alpha},c_n,\ldots,c_m,y_{m-1},\ldots,y_1,T_n,\ldots,T_{m+1},Y_m,\ldots,Y_1,T) \tag{28}$$

$$\forall y_m\ \exists Y_m\ \ldots\ \forall y_1\ \exists Y_1\ \theta(\vec{a},\vec{\alpha},c_n,\ldots,c_{m+1},y_m,\ldots,y_1,T_n,\ldots,T_{m+1},Y_m,\ldots,Y_1,T) \tag{29}$$

(c_i's are free number variables, and T, T_j's are string terms which do not involve bound variables y_i's and Y_j's). Also, the only quantifier introduction rules can be used in π are the number $\forall$–**right** rule and the string $\exists$–**right** rule.

We will prove by induction on the length of π that for each sequent $\mathcal{S}$ of π, there are functions F_i's of $\mathcal{L}$ (called the *witnessing functions* of $\mathcal{S}$) so that the sequent $\mathcal{S}'$, which is constructed from $\mathcal{S}$ and F_i's as described shortly, is a theorem of $\mathcal{T}$. Essentially F_i's are the witnessing functions that compute the existentially quantified string variables of $\mathcal{S}$, and $\mathcal{S}'$ is constructed from $\mathcal{S}$ by explicitly mentioning these witnessing functions. Suppose that $\mathcal{S} = \Lambda \longrightarrow \Gamma$ (note that Λ contains only open formulas), then $\mathcal{S}' = \Lambda \longrightarrow \Gamma'$, where Γ' consists of the following formulas. (We drop mention of $\vec{a},\vec{\alpha}$ in θ as well as in F_i's. Also F_i's may contain free variables that are present in $\mathcal{S}$.)

- All open formulas in Γ;
- For each formula of the form (28) in Γ, the formula

$$\theta(c_n,\ldots,c_m,b_{m-1},\ldots,b_1,T_n,\ldots,T_{m+1},F_m(c_n,\ldots,c_m),\ldots,F_1(c_n,\ldots,c_m,b_{m-1},\ldots,b_1),T) \tag{30}$$

- For each formula of the form (29) in Γ, the formula

$$\theta(c_n,\ldots,c_{m+1},b_m,\ldots,b_1,T_n,\ldots,T_{m+1},F_m(c_n,\ldots,c_{m+1},b_m),\ldots,F_1(c_n,\ldots,c_{m+1},b_m,\ldots,b_1),T) \tag{31}$$

[3] Anchored proofs are also called *free–cut free* proofs in the literature.

(In (30) and (31), the free variables b_i's do not appear anywhere else in $\mathcal{S}'$.)

The base case holds trivially, since the axioms of $\mathcal{T}$ are open formulas. For the induction step, we consider the inference rules that might be used in π.

Case I (String $\exists$–**right**): Suppose that $\mathcal{S}$ is the bottom sequent of the inference

$$\frac{\mathcal{S}_1}{\mathcal{S}} = \frac{\Lambda \longrightarrow \Gamma, \psi(T_{m+1})}{\Lambda \longrightarrow \Gamma, \exists Y_{m+1}\psi(Y_{m+1})}$$

where ψ is as (29). By the induction hypothesis, $\mathcal{S}_1'$ is a theorem of $\mathcal{T}$. We obtain $\mathcal{S}'$ from $\mathcal{S}_1'$ by taking F_{m+1} to be the function defined by T_{m+1}.

Case II (Number $\forall$–**right**): Note that this rule can be applied to only formulas of the form (28). Also in this case, the free variable c_m must not appear anywhere else in $\mathcal{S}$. No new function is required.

Case III (Cut): Note that the cut formula is an open formula. Suppose that $\mathcal{S}$ is derived from $\mathcal{S}_1$ and $\mathcal{S}_2$ using the cut rule:

$$\frac{\mathcal{S}_1 \qquad \mathcal{S}_2}{\mathcal{S}} = \frac{\Lambda \longrightarrow \Gamma, \psi \qquad \Lambda, \psi \longrightarrow \Gamma}{\Lambda \longrightarrow \Gamma}$$

where ψ is an open formula of $\mathcal{L}$. The witnessing functions of $\mathcal{S}$ is defined from the witnessing functions of $\mathcal{S}_1$ and $\mathcal{S}_2$ as follows:

$$F_i(z) \leftrightarrow (\neg\psi \wedge F_i^1(z)) \ \vee \ (\psi \wedge F_i^2(z))$$

Case IV (Weakening rule): If $\mathcal{S}$ is obtained from $\mathcal{S}_1$ by the weakening rule, then $\mathcal{S}'$ can be obtained from $\mathcal{S}_1'$ by the same rule. When the additional formula in $\mathcal{S}$ is a $\mathbf{g\Sigma}_1^B$ formula (of the form (28) or (29)), the witnessing functions can be the constant string function $\mathbf{0}$.

Case V (Contraction rule): Suppose that $\mathcal{S}$ is derived from $\mathcal{S}_1$ using the contraction rule. Consider the interesting case where the formula removed from $\mathcal{S}_1$ is a $\mathbf{g\Sigma}_1^B$ formula.

$$\frac{\mathcal{S}_1}{\mathcal{S}} = \frac{\Lambda \longrightarrow \Gamma, \psi(\vec{a},\vec{\alpha},\vec{c}), \psi(\vec{a},\vec{\alpha},\vec{c})}{\Lambda \longrightarrow \Gamma, \psi(\vec{a},\vec{\alpha},\vec{c})}$$

($\psi(\vec{a},\vec{\alpha},\vec{c})$ is of the form (28) or (29)). Note that the two occurrences of ψ in $\mathcal{S}_1$ may have different collections of witnessing functions in $\mathcal{S}_1'$. However, if $\forall\vec{z}\psi(\vec{a},\vec{\alpha},\vec{z})$ is to be true, then at least one of the two collections is correct. The witnessing functions in $\mathcal{S}'$ are defined using this information.

Formally, consider the case of (28), and assume that corresponding to the two occurrences of $\psi(\vec{a},\vec{\alpha},\vec{c})$ in $\mathcal{S}_1$, we have the following formulas in $\mathcal{S}_1'$ (see (30)):

$$\theta^1(\vec{c},\vec{b}) \equiv \theta(\vec{c},\vec{b},\vec{T},F_m^1(\vec{c}),\ldots,F_1^1(\vec{c},\vec{b}))$$
$$\theta^2(\vec{c},\vec{b}) \equiv \theta(\vec{c},\vec{b},\vec{T},F_m^2(\vec{c}),\ldots,F_1^2(\vec{c},\vec{b}))$$

In general, the witnessing functions of $\mathcal{S}'$ are

$$F_i(\vec{c},\vec{b})(x) \leftrightarrow (\forall\vec{z}\,\forall\vec{y}\,\theta^1(\vec{z},\vec{y}) \wedge F_i^1(\vec{c},\vec{b})(x)) \ \vee \ (\neg\forall\vec{z}\,\forall\vec{y}\,\theta^1(\vec{z},\vec{y}) \wedge F_i^2(\vec{c},\vec{b})(x))$$

Case VI (Other rules) The introduction rules for $\wedge, \vee, \neg$ can be applied to only open formulas. No new function is required. Similarly, if $\mathcal{S}$ is obtained from $\mathcal{S}_1$ by the exchange rule, then $\mathcal{S}'$ can be derived from $\mathcal{S}_1'$ by the same rule. $\square$

Corollary 3.21 (Witnessing Theorems for $\mathbf{VTC}^0$). *For each theorem $\exists\vec{Z}\varphi(\vec{x},\vec{X},\vec{Z})$ of $\mathbf{VTC}^0$ where φ is $\mathbf{g\Sigma}_1^B$, there are string functions $\vec{F} \in \mathbf{FTC}^0$, such that*

$$\mathbf{VTC}^0 \cup \mathrm{AX}(\mathcal{L}(\vec{F})) \vdash \varphi(\vec{x},\vec{X},\vec{F}(\vec{x},\vec{X})).$$

Proof. Suppose that $\mathbf{VTC}^0 \vdash \exists \vec{Z}\varphi(\vec{x},\vec{X},\vec{Z})$, so $\overline{\mathbf{VTC}}^0 \vdash \exists \vec{Z}\varphi(\vec{x},\vec{X},\vec{Z})$. By Theorem 3.20, there are string functions $\vec{F} \in \mathcal{L}_{\mathbf{FTC}^0}$ such that $\overline{\mathbf{VTC}}^0 \vdash \varphi(\vec{x},\vec{X},\vec{F}(\vec{x},\vec{X}))$. The conclusion follows from Corollary 3.19. □

Note that similar witnessing theorems hold for the universal theory $\overline{\mathbf{VTC}}^0$.

Corollary 3.22 (The remaining direction of Theorem 3.6). *The $\mathbf{\Sigma}_1^1$-definable function in* $\mathbf{VTC}^0$ *are in* $\mathbf{FTC}^0$.

3.6 Theories for Other Subclasses of P

In this subsection, we will apply Theorem 3.15 and Theorem 3.20 to develop finitely axiomatizable theories for other uniform subclasses of $\mathbf{P}$ in the same style as $\mathbf{VTC}^0$. Let F be a polynomial time string function and let $\mathbf{C}$ be the class of two–sorted relations which are $\mathbf{AC}^0$–reducible to F. Note that the associated function class $\mathbf{FC}$ is usually defined using the graphs (for number functions) and bit graphs (for string functions) (page 5). But it can be equivalently defined as the class of functions which are $\mathbf{AC}^0$–reducible to F. In the case of $\mathbf{TC}^0$, F is essentially the string function computing the "counting array", whose graph is given by *NUMONES*.

We add to $\mathbf{V}^0$ a $\mathbf{\Sigma}_1^B$ axiom $AXIOM_F$ which formalizes the polytime algorithm that computes F. (Thus $AXIOM_F$ is a generalization of *NUMONES*.) We will show that the resulting theory $\mathcal{T}$ characterizes $\mathbf{C}$. The proof is almost identical to the proof in the case of $\mathbf{VTC}^0$: we show that the universal theory $\overline{\mathcal{T}}$ (obtained from $\mathcal{L}_A^2(Row, F)$ in the same way that $\overline{\mathbf{VTC}}^0$ is obtained from $\mathcal{L}_A^2(Row, numones)$) characterizes the same class. The main task is to prove the analogue of Lemma 3.16, i.e., $\mathcal{T}(Row, F)$ satisfies hypotheses a) and b) in Theorem 3.15. Our choice of the axiom $AXIOM_F$ will make this step readily obtainable. [4]

The universal defining axiom for F is obtained from a modified version of Cobham's recursion theoretic characterization of the polytime functions. Here we use the fact that each polytime function can be obtained from $\mathbf{AC}^0$ functions by composition and at most one application of the *bounded recursion* operation. In each complexity class of interest it turns out that a suitable function F complete for the class can be defined by such a recursion of the form

$$F(0, X) = (Init(X))^{<t(0,|X|)}$$
$$F(x+1, X) = (Next(x, X, F(x, X)))^{<t(x+1,|X|)}$$

where $Init(X)$ and $Next(x, X, Y)$ are $\mathbf{AC}^0$ functions, $t(x, y)$ is a polynomial, and $X^{<y}$ is the initial segment of X of length y.

We will first define the universal theory $\overline{\mathcal{T}}$ in the same manner that we have defined $\overline{\mathbf{VTC}}^0$. Here we will not introduce the new functions $Init, Next$ and $X^{<y}$ but will use their $\mathbf{\Sigma}_0^B$ definitions instead. In other words, F can be defined as follows:

$$F(0, X)(z) \leftrightarrow z < t(0, |X|) \wedge \varphi_{Init}(z, X) \tag{32}$$
$$F(x+1, X)(z) \leftrightarrow z < t(x+1, |X|) \wedge \varphi_{Next}(z, x, X, F(x, X)) \tag{33}$$

where φ_{Init} and φ_{Next} are the $\mathbf{\Sigma}_0^B$ bit definitions of *Init* and *Next* respectively. (Note that in $\overline{\mathbf{V}}^0$, φ_{Init} and φ_{Next} are equivalent to open formulas of $\mathcal{L}_{\mathbf{FAC}^0}$.)

The language $\mathcal{L}_{\mathbf{FC}}$ of functions in $\mathbf{FC}$ is defined in the same way as $\mathcal{L}_{\mathbf{FTC}^0}$ (Definition 3.10), except for *numones* is replaced by F. The theory $\overline{\mathcal{T}}$ is defined similarly to $\overline{\mathbf{VTC}}^0$ (Definition 3.12), with the defining

[4] Note the fact that $\mathbf{\Sigma}_1^1$ theorems of $\mathcal{T}$ can be witnessed by functions of $\mathbf{FC}$ follows easily from Herbrand's Theorem. The General Witnessing Theorem offers more than we need here. It is necessary in Section 4 where we show that $\mathbf{VTC}^0$ is RSUV isomorphic to $\mathbf{\Delta}_1^b$-$\mathbf{CR}$.

axioms (32) and (33) of F replacing the defining axioms of *numones*. The following Corollary follows from Theorem 3.20.

Corollary 3.23. *For each theorem $\exists\vec{Z}\varphi(\vec{a},\vec{\alpha},\vec{Z})$ of $\overline{\mathcal{T}}$, where φ is a $\mathbf{g\Sigma}_1^B$ formula, there are functions $\vec{F}$ of $\mathcal{L}_{\mathbf{FC}}$ such that*

$$\overline{\mathcal{T}} \vdash \forall\vec{x}\forall\vec{X}\ \varphi(\vec{x},\vec{X},\vec{F}(\vec{x},\vec{X})).$$

On the other hand, $\overline{\mathcal{T}}$ proves the $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FC}})$-**COMP** scheme, and thus can $\mathbf{\Sigma}_1^B(\mathcal{L}_{\mathbf{FC}})$–define all functions of $\mathcal{L}_{\mathbf{FC}}$.

Now we will define $\mathcal{T}$. Our choice for the $\mathbf{\Sigma}_1^B$ defining axiom of F comes from the above definition of F given in (32), (33). In order to prove Claim 3.26 (see the discussion below) we will not compute F for a single value of X, but rather multiple (i.e., polynomially many) values of X. Let $\varphi_F(a,b,X,Y)$ be the formula stating that Y encodes simultaneously the b recursive computations of $F(a,X^{[0]}),\ldots,F(a,X^{[b-1]})$ using the definition of F. More precisely, let $\varphi_F(a,b,X,Y)$ be

$$\begin{aligned}&\forall y < b,\ [\forall z < aY^{[y,0]}(z) \leftrightarrow \varphi_{Init}(z,X^{[y]}) \wedge \\ &\qquad \forall x < a\forall z < a,\ Y^{[y,x+1]}(z) \leftrightarrow z < t(x+1,|X^{[y]}|) \wedge \varphi_{Next}(z,x,X,Y^{[y,x]})]\end{aligned} \tag{34}$$

Definition 3.24. *Let $AXIOM_F$ be $\forall a\forall b\forall X\exists Y \le \langle b, t(a,|X|)\rangle\varphi_F(a,b,X,Y)$. The theory $\mathcal{T}$ is $\mathbf{V}^0$ extended by the axiom $AXIOM_F$.*

Since $\mathbf{V}^0$ is finitely axiomatizable, so is $\mathcal{T}$.

Lemma 3.25. *$\overline{\mathcal{T}}$ is a conservative extension of $\mathcal{T}$ and satisfies the hypotheses a), b) of Theorem 3.15.*

Proof. The proof is the same as the first part of the proof of Corollary 3.18. First, let $\mathcal{T}(Row,F)$ be $\mathcal{T}$ together with the functions *Row* and F and their defining axioms (5), (32), (33). Note that $\mathcal{T}(Row,F)$ is conservative over $\mathcal{T}$, because both *Row* and F are $\mathbf{\Sigma}_1^B$-definable in $\mathcal{T}$. (The $\mathbf{\Sigma}_1^B$-definability of F follows by $AXIOM_F$.) Assume that

Claim 3.26. *$\mathcal{T}(Row,F)$ satisfies hypotheses a) and b) in Theorem 3.15.*

Then $\overline{\mathcal{T}}$ is obtained from $\mathcal{T}(Row,F)$ by a series of conservative extensions satisfying hypotheses a) and b) of Theorem 3.15.

It remains to prove the Claim. We proceed as in the proof of Theorem 3.15. In fact, it suffices to show that $\mathcal{T}(Row,F)$ proves the existence of W such that for all $\vec{z} < b$, $W^{[\vec{z}]}$ is the intended value of $F(s,T)$ where s,T are terms of $\mathcal{L}_A^2(Row)$ which may contain $\vec{z}$. (Compare to (26).) Using $AXIOM_F$, such W can be constructed using $\mathbf{\Sigma}_0^B$-**COMP**. □

Corollary 3.27. *The $\mathbf{\Sigma}_1^1$–definable (and the $\mathbf{\Sigma}_1^B$–definable) functions in $\mathcal{T}$ are precisely those in $\mathcal{L}_{\mathbf{FC}}$.*

Proof. Each function of $\mathcal{L}_{\mathbf{FC}}$ has a $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FC}})$ definition, which is equivalent in $\overline{\mathcal{T}}$ to a $\mathbf{\Sigma}_1^B$ formula, by Lemma 3.25. It is therefore $\mathbf{\Sigma}_1^B$ definable in $\mathcal{T}$.

On the other hand, $\mathbf{\Sigma}_1^1$ theorems of $\overline{\mathcal{T}}$ (and hence of $\mathcal{T}$) are witnessed by **FC** functions, as shown in Corollary 3.23. □

Note that the axiom *NUMONES* is a special case of $AXIOM_F$. It is "nicer" than $AXIOM_F$ in the sense that it encodes only a single computation of *numones*. In fact, Claim 3.17 shows that $\mathbf{VTC}^0(Row)$ proves $AXIOM_{numones}$. We need this in order to show that $\mathbf{VTC}^0$ satisfies the hypotheses a) and b) of Theorem 3.15 (Lemma 3.16). However, the proof of this Claim is rather *ad hoc*.

In general, our choice of $AXIOM_F$ guarantees that $\mathcal{T}(Row, F)$ satisfies the hypotheses a) and b) of Theorem 3.15 (as shown in Claim 3.26). Thus to go further and obtain "nicer" axiom than $AXIOM_F$ (in the style of *NUMONES*), it remains to prove the analogue of Claim 3.17. These proofs may differ for different chosen functions F. Some examples are given below.

3.6.1 Theories for $\mathbf{AC}^0(m)$ and ACC

Theories for the complexity classes $\mathbf{AC}^0(m)$ and **ACC** are defined in the same way that $\mathbf{VTC}^0$ is defined for the class $\mathbf{TC}^0$. For $m \geq 2$, $\varphi_{MOD_m}(X, Y)$ is the formula stating that Y is the "counting modulo m" array for X:

$$\begin{aligned}\varphi_{MOD_m}(X,Y) \equiv & [\forall z \leq |X| \exists! y < m Y(z,y)] \ \wedge \ Y(0,0) \ \wedge \ \forall z < |X| \forall y < m, \\ & Y(z,y) \supset [(X(z) \supset Y(z+1, \ y+1 \mod m)) \wedge (\neg X(z) \supset Y(z+1,y))].\end{aligned} \tag{35}$$

Here, we identify the natural number m with the corresponding numeral $\underline{m}$. We take $\varphi(y \mod m)$ as an abbreviation for

$$\exists r < m, \exists q \leq y, \ y = qm + r \wedge \varphi(r). \tag{36}$$

Thus if $\varphi(y)$ is $\mathbf{\Sigma}_0^B$, then $\varphi(y \mod m)$ is also $\mathbf{\Sigma}_0^B$.

Definition 3.28. *For each $m \geq 2$, let $MOD_m \equiv \forall X \exists Y \varphi_{MOD_m}(X, Y)$. Then*

$$\begin{aligned}\mathbf{V}^0(m) &= \mathbf{V}^0 \cup \{MOD_m\} \\ \mathbf{VACC} &= \mathbf{V}^0 \cup \{MOD_m \mid m \geq 2\}.\end{aligned}$$

Note that the string Y in MOD_m can be bounded by $\langle |X|, m\rangle$.

The following Theorem can be proved in the same way as Theorem 3.6:

Theorem 3.29. *The Σ_1^1-definable (and the Σ_1^B-definable) functions in $\mathbf{V}^0(m)$, and* **VACC** *are precisely those in $\mathbf{FAC}^0(m)$, and* **FACC***, respectively.*

Corollary 3.30. *If* **VACC** *is finitely axiomatizable, then $\mathbf{ACC} = \mathbf{AC}^0(m)$, for some m.*

Proof. If **VACC** is finitely axiomatizable, then by compactness, it is equal to

$$\mathbf{V}^0 \cup \{MOD_i \mid 2 \leq i \leq m'\},$$

for some m'. Let $m = lcm\{2, \ldots, m'\}$, then

$$\mathbf{V}^0(m) \vdash \{MOD_i \mid 2 \leq i \leq m'\}.$$

Therefore $\mathbf{VACC} = \mathbf{V}^0(m)$, and the conclusion follows from the theorem. □

Corollary 3.31. *If* $\mathbf{VACC} \vdash NUMONES$*, then $\mathbf{TC}^0 = \mathbf{AC}^0(m)$, for some m.*

3.6.2 Theories for $\mathbf{NC}^k$ and NC

A language is in nonuniform $\mathbf{NC}^1$ if it is computable by a polynomial–size log–depth family of Boolean circuits. Here we use uniform $\mathbf{NC}^1$, which means **Alogtime**. Buss [5] shows that the Boolean formula value problem is complete for **Alogtime**. In general, for each $k \in \mathbb{N}$ uniform $\mathbf{NC}^k$ can be considered as the class of relations which are $\mathbf{AC}^0$–reducible to the circuit value problem, where the depth of the circuit is bounded by $(\log n)^k$. The function class $\mathbf{FNC}^k$ consists of functions $\mathbf{AC}^0$–reducible to the above problem, or equivalently the functions computable by uniform polynomial–size $(\log n)^k$–depth constant–fanins families of Boolean circuits. Also,

$$\mathbf{NC} = \bigcup_{k\geq 1} \mathbf{NC}^k, \qquad \mathbf{FNC} = \bigcup_{k\geq 1} \mathbf{FNC}^k$$

The two–sorted theory $\mathbf{VNC}^1$ introduced in [11, 14] is originated from Arai's single–sorted theory **AID** [1]. It is the theory $\mathbf{V}^0$ extended by the axiom scheme $\mathbf{\Sigma}_1^B$–*TreeRec*, which essentially exhibits the evaluations of log–depth Boolean circuits given their specification and inputs.

Informally, consider a log–depth Boolean circuit (i.e., a formula) whose gates can be numbered such that the input gates are numbered $a, \ldots, 2a-1$, output gate numbered 1 and other internal gates are numbered $2, \ldots, a-1$. Furthermore, inputs to gate i (where $i < a$) are from gates numbered $2i$ and $2i+1$. Let the gate i be given by a $\mathbf{\Sigma}_0^B$ formula $\phi(i)[p,q]$ which might have other parameters, i.e., the intended meaning of $\phi(i)[p,q]$ is the output of the gate numbered i when its two inputs are p, q. The $\mathbf{\Sigma}_1^B$–*TreeRec* for ϕ explicitly evaluates all the gates of such circuit when it is given inputs $Z(0), \ldots Z(a)$: for $i < a$, $Z(i)$ is the value output by gate numbered i. Formally, it is defined as follows:

$$\exists Z \leq 2a \forall i < a[Z(i+a) \leftrightarrow \psi(i) \wedge 0 < i \supset (Z(i) \leftrightarrow \phi(i)[Z(2i), Z(2i+1)]]$$

It has been shown [11, 14] that the $\mathbf{\Sigma}_1^1$–definable functions in $\mathbf{VNC}^1$ are precisely the functions in $\mathbf{FNC}^1$, the function class associated with $\mathbf{NC}^1$. It has also been shown that $\mathbf{VNC}^1$ can be axiomatized by $\mathbf{V}^0$ and the axiom *TE*. Here *TE* is an instance of $\mathbf{\Sigma}_1^B$–*TreeRec*, where ϕ^{TE} is used in place of ϕ, is either $\wedge$ or $\vee$

$$\phi^{TE}(i)[p,q] \equiv (\mathit{parity}(\log i) \wedge (p \vee q)) \wedge (\neg \mathit{parity}(\log i) \wedge (p \wedge q))$$

Loosely speaking, each gate in the log–depth circuit specified by ϕ^{TE} depending on the parity of its *depth* in the circuit. [5]

We will now define the theories characterizing $\mathbf{NC}^k$ (note that for $k = 1$ we obtain the same theory as $\mathbf{VNC}^1$, but we will not prove this fact here). For each k, the complete problem for $\mathbf{NC}^k$ is given by a circuit of depth $O((\log n)^k)$ and its inputs. Without loss of generality, consider a circuit of depth $(\log a)^k$, where each layer contains at most a gates. Furthermore, assume that the layers alternate between $\wedge$ and $\vee$, and are indexed according to their depths: 0 (input gates), …, $(\log a)^k$ (output gates), with the outputs of gates on layer d connect to the inputs of gates on layer $d+1$. On each layer, the gates are numbered $0, \ldots, a$ (i.e., any gate is indexed by its layer and its position on the layer).

Such circuit can be described by listing the gates together with their layers index and their inputs gates positions (on the layer below it). Thus we will have a string variable Y where for $d < \log^k a$ and $x, y, z < a$, $Y^{[d]}(x,y,z)$ holds if and only if inputs to gate z on layer $d+1$ are from gates x, y on layer d. The formula $\varphi_{k-CE}(a, X, Y, Z)$ below states that Z evaluates all the gates of the circuit specified by Y when it is given inputs $X(0), \ldots, X(a-1)$. In particular, the output of gate z on layer d is $Z^{[d]}(z)$.

$$\begin{aligned}\varphi_{k-CE}(a,X,Y,Z) \equiv \forall d < \log^k a \forall z < a \exists! x, y < a Y^{[d]}(x,y,z) \supset &\\ \forall z < a Z^{[0]}(z) \leftrightarrow X(z) \wedge &\\ \forall d < \log^k a \forall x,y,z < a,\ Y^{[d]}(x,y,z) \supset (Z^{[d+1]}(z) \leftrightarrow B(d, Z^{[d]}(x), Z^{[d]}(y))) &\end{aligned} \tag{37}$$

where $B(d,p,q)$ is

$$(\mathit{parity}(d) \wedge (p \vee q)) \vee (\neg \mathit{parity}(d) \wedge (p \wedge q)).$$

Definition 3.32. *Let k–CE denote* $\exists Z \varphi_{k-CE}(a,X,Y,Z)$*. The theory* $\mathbf{VNC}^k$ *is* $\mathbf{V}^0$ *extended by the axiom k–CE. The theory* $\mathbf{VNC}$ *is*

$$\bigcup_{k\geq 1} \mathbf{VNC}^k$$

[5]Note that $\log i$ is definable in $\mathbf{I\Delta}_0$, hence in $\mathbf{V}^0$ [10].

Again, note that in k–CE Z can be bounded, therefore $\mathbf{VNC}^k$ is equivalent to a theory with bounded axioms. Proving the first sentence in Theorem 3.33 below is somewhat easier than Claim 3.17.

Theorem 3.33. *For each $k \geq 1$,*

$$\mathbf{VNC}^k \vdash \forall X \forall Y \exists Z \forall w < b \varphi_{k-CE}(a, X^{[w]}, Y^{[w]}, Z^{[w]}).$$

A function is in $\mathbf{FNC}^k$ iff it is $\mathbf{\Sigma}_1^1$–definable in $\mathbf{VNC}^k$. A function is in $\mathbf{FNC}$ iff it is $\mathbf{\Sigma}_1^1$–definable in $\mathbf{VNC}$.

3.6.3 Theories for NL, SL, L and P

$\mathbf{NL}$ is the class of problems solvable in a nondeterministic Turing machine in space $O(\log n)$. We consider $\mathbf{NL}$ as the class of two–sorted relations $\mathbf{AC}^0$–reducible to the *Graph Accessibility Problem* GAP (also known as *Path*, or *Reachability* problem). This is the problem of deciding whether there is a path from s to t in a given (directed) graph G, where s, t are the 2 designated vertices of G.

We can obtain a theory that characterizes $\mathbf{NL}$ by formalizing the following polytime algorithm that solves GAP. For each distance $k = 0, 1, \ldots, n-1$ (where n is the number of vertices in the graph), simply list all vertices that can be reached from s by paths of length at most k. This enables us to check if t is reachable from s by paths of length at most n, i.e., if there is a path from s to t in G.

Using GAP, the theory $\mathbf{VNL}$ is developed in the same style of $\mathbf{VTC}^0$. The $\mathbf{\Sigma}_1^B$ axiom that formalizes the algorithm solving this problem is called LC (for Logspace Computation). In the following definition, E codes a directed graph, and Z is intended to code the above polytime algorithm. Here we identify the source s with 0. Let φ_{LC} be the following formula

$$\begin{aligned}\varphi_{LC}(a, E, Z) \equiv\ & Z(0,0) \wedge \forall i < a \neg Z(0,i) \wedge \\ & \forall k, i < a,\ Z(k+1, i) \leftrightarrow [Z(k,i) \vee \exists j < a,\ E(j,i) \wedge Z(k,j)].\end{aligned} \tag{38}$$

Note that in (38), $Z(k,i)$ holds iff there is a path from 0 to i of length at most k.

Definition 3.34 (VNL). *Let LC denote $\forall a \forall E \exists Z \leq \langle a, a\rangle \varphi_{LC}(a, E, Z)$, Then $\mathbf{VNL}$ is the theory $\mathbf{V}^0$ extended by the axiom LC.*

It can be shown directly that the class of $\mathbf{\Sigma}_1^B$–definable functions in $\mathbf{VNL}$ is precisely $\mathbf{FNL}$, the class of functions whose bit graphs are in $\mathbf{NL}$. Here we can show this using Corollary 3.27. It amounts to showing that $\mathbf{VNL}$ proves the axiom $AXIOM_{F_{GAP}}$ (where $F_{GAP}(a, E)$ is essentially the function whose graph is $\varphi_{LC}(e, E, Z)$), i.e.,

$$\mathbf{VNL} \vdash \forall a \forall b \forall E \exists Z \leq \langle b, a, a\rangle \forall y < b \varphi_{F_{GAP}}(a, E^{[y]}, Z^{[y]}).$$

This is analogous to Claim 3.17. The proof idea is similar; details are omitted.

In the same spirit, a series of theories for $\mathbf{L}$ (class of problems solvable by a Turing machine in space $O(\log n)$), $\mathbf{SL}$ (class of problems solvable by a *symmetric nondeterministic* Turing machine in space $O(\log n)$), and $\mathbf{P}$ can be obtained using similar complete problems. For $\mathbf{L}$ the complete problem is GAP restricted to directed graphs whose vertices have out degree at most 1; for $\mathbf{SL}$ the complete problem is GAP restricted to undirected graphs; and for $\mathbf{P}$ the complete problem is the circuit value problem.

Remark It is not a surprise that the theories obtained this way are "minimal", and thus coincide with a number of existing "minimal" theories that characterize the corresponding classes. In fact, it can be shown that in case of $\mathbf{L}$, the theory obtained is actually Zambella's theory $\mathbf{\Sigma}_0^B$-**Rec** [31], and in case of $\mathbf{P}$, the theory obtained is the same as $\mathbf{TV}^0$ [11] (and thus the same as $\mathbf{V}^1$-**HORN** [12]). Thus, these results explicitly exhibit the finite axiomatizability of $\mathbf{\Sigma}_0^B$-**Rec** and $\mathbf{TV}^0$. In the case of $\mathbf{NL}$, it has been shown [22] that $\mathbf{VNL}$ is the same as $\mathbf{V}^1$-**KROM** (see also [13]). In the next section we will show that $\mathbf{VTC}^0$ is RSUV isomorphic to Johannsen and Pollett's "minimal" theory $\mathbf{\Delta}_1^b$-**CR**.

4 RSUV Isomorphism Between $\mathbf{VTC}^0$ And $\mathbf{\Delta}_1^b$-CR

4.1 The Theory $\mathbf{\Delta}_1^b$-CR

The theory $\mathbf{\Delta}_1^b$-**CR** [21] is a single–sorted theory whose $\mathbf{\Sigma}_1^b$ definable functions are precisely the (single–sorted) $\mathbf{TC}^0$ functions. It is claimed to be a "minimal" theory for $\mathbf{TC}^0$. We will show that it is *RSUV isomorphic* to our theory $\mathbf{VTC}^0$. First, we recall the definition of $\mathbf{\Delta}_1^b$-**CR**.

The underlying vocabulary of $\mathbf{\Delta}_1^b$-**CR** is

$$\mathcal{L}_{\mathbf{\Delta}_1^b\text{-}\mathbf{CR}} = [0, S, +, \cdot, \lfloor \tfrac{1}{2}x \rfloor, |x|, x\#y, \dot{-}, MSP; \leq]$$

(here S is the successor function, and MSP stands for most significant bits, $MSP(x,i) = \lfloor x/2^i \rfloor$). The theory $\mathbf{\Delta}_1^b$-**CR** is axiomatized by the defining axioms for symbols of $\mathcal{L}_{\mathbf{\Delta}_1^b\text{-}\mathbf{CR}}$, the axiom scheme **Open-LIND**, and the $\mathbf{\Delta}_1^b$ bit–comprehension rule (below).

The defining axioms of the symbols of $\mathcal{L}_{\mathbf{\Delta}_1^b\text{-}\mathbf{CR}}$ are straightforward. The axiom scheme **Open-LIND** can be seen as a scheme of induction on "small" numbers (i.e., $|z|$) for quantifier-free formulas. Formally, **Open-LIND** is the set of

$$[\varphi(0) \wedge \forall x,\ \varphi(x) \supset \varphi(Sx)] \supset \forall z \varphi(|z|), \tag{39}$$

where φ is an open formula. The $\mathbf{\Delta}_1^b$ bit–comprehension rule is defined as follows. First, given a formula $\varphi(i)$ (which might have other free variables), the comprehension axiom for $\varphi(i)$, denoted by $\mathbf{COMP}_{\varphi(i)}(a)$, is the formula

$$\exists x < 2^{|a|} \forall i < |a|\ [BIT(i,x) \leftrightarrow \varphi(i)].$$

Here $BIT(i,x)$ holds if and only if the ith bit in the binary representation of x is 1 (the bits of x are counted from 0 for the lowest order bit). It is defined by

$$BIT(i,x) \equiv mod2(MSP(x,i)), \qquad \text{where } mod2(x) = x \dot{-} 2 \cdot \lfloor \tfrac{1}{2}x \rfloor.$$

Then, the $\mathbf{\Delta}_1^b$ bit–comprehension rule is the following inference rule:

$$\frac{\varphi(i) \leftrightarrow \psi(i)}{\mathbf{COMP}_{\varphi(i)}(t)}$$

where φ is a $\mathbf{\Sigma}_1^b$ formula, ψ is a $\mathbf{\Pi}_1^b$ formula, and t is a term.

Note that formally, $\mathbf{\Delta}_1^b$-**CR** is defined inductively using the above rule. More precisely, $\mathbf{\Delta}_1^b$-**CR** is the smallest theory that contains the axioms described above, and is closed under the $\mathbf{\Delta}_1^b$ bit–comprehension rule, i.e., if $\varphi(i) \leftrightarrow \psi(i)$ is in $\mathbf{\Delta}_1^b$-**CR** for some $\mathbf{\Sigma}_1^b$ formula φ and $\mathbf{\Pi}_1^b$ formula ψ, then $\mathbf{COMP}_{\varphi(i)}(t)$ is also in $\mathbf{\Delta}_1^b$-**CR**, for any term t. Let $\mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ be the subtheory of $\mathbf{\Delta}_1^b$-**CR** where proving each theorem of $\mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ requires at most i applications of the $\mathbf{\Delta}_1^b$ bit–comprehension rule. Then

$$\mathbf{\Delta}_1^b\text{-}\mathbf{CR} = \bigcup_{i \geq 0} \mathbf{\Delta}_1^b\text{-}\mathbf{CR}_i.$$

An open question [21] is whether $\mathbf{\Delta}_1^b$-**CR** $=$ $\mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ for some constant i. Since $\mathbf{VTC}^0$ is finitely axiomatizable, the RSUV isomorphism between $\mathbf{\Delta}_1^b$-**CR** and $\mathbf{VTC}^0$ proved below shows that $\mathbf{\Delta}_1^b$-**CR** is also finitely axiomatizable. It follows that $\mathbf{\Delta}_1^b$-**CR** $=$ $\mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ for some constant i.

Note that there is no side formula in the $\mathbf{\Delta}_1^b$ bit comprehension rule, and thus it is apparently weaker than the $\mathbf{\Delta}_1^b$ bit–comprehension axiom scheme:

$$\forall i(\varphi(i) \leftrightarrow \psi(i)) \supset \mathbf{COMP}_{\varphi(i)}(t) \tag{40}$$

where φ is a $\mathbf{\Sigma}_1^b$ formula and ψ is a $\mathbf{\Pi}_1^b$ formula. In fact, [15] shows that $\mathbf{\Delta}_1^b$-**CR** does not prove the above comprehension axiom scheme unless RSA can be cracked using probabilistic polynomial time algorithms.

4.2 RSUV Isomorphism

We will prove that $\mathbf{\Delta}_1^b$-**CR** is RSUV isomorphic [23, 28, 29] to our theory $\mathbf{VTC}^0$. A major part of this proof is in defining multiplication for second sort objects and proving the commutative and distributive laws. Here we identify each bounded subset X with the number

$$X(0)2^0 + \ldots X(n-1)2^{n-1},$$

where $n = |X|$. Then the multiplication function $X \cdot Y$ is defined as the product of these "big numbers" corresponding to X and Y. Note that it is quite straightforward to formalize the conventional (polynomial time) algorithm for multiplication in theories that characterize $\mathbf{P}$, such as $\mathbf{V}^1$ [10]. However this is less straightforward in the case of $\mathbf{VTC}^0$. Indeed, the conventional algorithm might not be in $\mathbf{TC}^0$. Note that the multiplication function is already complete for $\mathbf{TC}^0$. Here we define this function in $\mathbf{VTC}^0$ using the fact that computing Y from X in (6) (or equivalently the function *numones*) is also complete for $\mathbf{TC}^0$. In particular, we define multiplication in $\mathbf{VTC}^0$ by formalizing a reduction from *numones*. We use the same method as in [3], where it is shown that the properties of multiplication have small $\mathbf{TC}^0$-**Frege** proofs.

Note that reasoning in $\overline{\mathbf{VTC}}^0$ is more uniform than in $\mathbf{TC}^0$-**Frege**. For example, in $\overline{\mathbf{VTC}}^0$ the "input bits" are already ordered (i.e., the bits in the string $X(n-1)\ldots X(0)$ are numbered), while this is not the case in $\mathbf{TC}^0$-**Frege**. Consequently, the definition of the sum of n strings in $\mathbf{TC}^0$-**Frege** does not depend on the order of the strings (i.e., it is symmetric in terms of the arguments). In $\mathbf{VTC}^0$ proving this independence is a nontrivial task. It is true, although nontrivial, that each $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FTC}^0})$ theorem of $\overline{\mathbf{VTC}}^0$ translates into a family of propositional tautologies having polynomial size d-**PTK** proofs (for some constant d depending on the theorem). Together with the results here, it follows that for some constant d, d-**PTK** can define string multiplication and prove its properties using polynomial size proofs. Our d-**PTK** proofs here can be seen as uniform versions of the $\mathbf{TC}^0$-**Frege** proofs in [3].

4.2.1 Outline of the RSUV Isomorphism

We prove the RSUV isomorphism between $\mathbf{\Delta}_1^b$-**CR** and $\mathbf{VTC}^0$ by (a) constructing from each model $\mathcal{M}$ of $\mathbf{\Delta}_1^b$-**CR** a model $\mathcal{N}$ of $\mathbf{VTC}^0$ whose second sort universe is the universe M of $\mathcal{M}$, and whose first sort universe if the subset $\log(M) = \{|u| \mid u \in M\}$; and (b) constructing from each model $\mathcal{N}$ of $\mathbf{VTC}^0$ a model $\mathcal{M}$ of $\mathbf{\Delta}_1^b$-**CR** whose universe if the second sort universe of $\mathcal{N}$. These constructions have the property that if we follow (a) to get a model $\mathcal{N}$ of $\mathbf{VTC}^0$ from the model $\mathcal{M}$ of $\mathbf{\Delta}_1^b$-**CR**, and then follow (b) to get a model $\mathcal{M}'$ of $\mathbf{\Delta}_1^b$-**CR** from $\mathcal{N}$, then $\mathcal{M}'$ and $\mathcal{M}$ are isomorphic. Similarly, if we start with a model $\mathcal{N}$ of $\mathbf{VTC}^0$ and follow (b) to get a model $\mathcal{M}$ of $\mathbf{\Delta}_1^b$-**CR**, then follow (a) to get $\mathcal{N}'$ from $\mathcal{M}$, then $\mathcal{N}$ and $\mathcal{N}'$ are isomorphic.

Corresponding to (a) there is a syntactic translation sending a sentence φ in the language of $\mathbf{VTC}^0$ to an equivalent sentence $\varphi^\flat$ in the language of $\mathbf{\Delta}_1^b$-**CR** and corresponding to (b) there is a syntactic translation sending a sentence ψ of $\mathbf{\Delta}_1^b$-**CR** to an equivalent sentence $\psi^\sharp$ in the language of $\overline{\mathbf{VTC}}^0$. (The $\flat$ and $\sharp$ notation is from [28].) Recall from Corollary 3.18 that $\overline{\mathbf{VTC}}^0$ is a conservative extension of $\mathbf{VTC}^0$. These translations can be pictured as follows:

$$\begin{array}{ccc}
\mathbf{\Delta}_1^b\text{-}\mathbf{CR} & \simeq & \mathbf{VTC}^0 \\
\mathcal{M} & \rightharpoonup & \mathcal{N} \\
\varphi^\flat & \leftharpoondown & \varphi \\
\hline
\mathcal{M}' & \leftharpoondown & \mathcal{N}' \\
\psi & \rightharpoonup & \psi^\sharp
\end{array}$$

The construction in (a) is straightforward. Let $\mathcal{M}$ be a model of $\mathbf{\Delta}_1^b$-**CR** with universe M, we construct a model $\mathcal{N}$ of $\mathbf{VTC}^0$ as follows. To get the second sort universe of $\mathcal{N}$, we simply identify each number $a \in M$

with the subset X (bounded by $|a|$) of those i such that the ith bit in the binary representation of a is 1. The symbols of $\mathbf{VTC}^0$ are interpreted accordingly, i.e, $0, 1, +, \cdot, =_1$ and $\leq$ are interpreted as in $\mathcal{M}$ (restricted to $\log(M)$), and

$$|X| = |a|, \quad \text{and } i \in X \Leftrightarrow BIT(i,a) = 1 \qquad \text{if } X = \{i \mid BIT(i,a) = 1\},$$
$$X =_2 Y \text{ if they are mapped from the same number } a \in M.$$

It remains to show that the axioms of $\mathbf{VTC}^0$ hold in $\mathcal{N}$. We use the fact that each $\mathbf{\Sigma}_0^B$ formula φ of $\mathbf{VTC}^0$ translates to the formula φ^b which is provably equivalent in $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ to a $\mathbf{\Sigma}_0^b$ formula. Since $\mathcal{M}$ is a model of $\mathbf{\Delta}_1^b$-$\mathbf{CR}$, it is easy to check that the axioms in 2-$\mathbf{BASIC}$ are satisfied in $\mathcal{N}$. The axiom scheme $\mathbf{\Sigma}_0^B$-$\mathbf{COMP}$ are satisfied in $\mathcal{N}$ since $\mathcal{M}$ satisfies the $\mathbf{\Delta}_1^b$ bit–comprehension rule. Now we show that $NUMONES$ holds in $\mathcal{N}$. Let $a \in M$, and $a_{n-1} \ldots a_0$ be the binary representation of a. We need to get the "counting array" for the set

$$X = \{i \mid BIT(i,a) = 1\}.$$

Let $a' \in M$ whose binary representation is $a_{n-1}0\ldots0a_{n-2}0\ldots0a_0$, where every block of 0's has length $(1 + |n|)$. Let $b \in M$ with the binary representation $10\ldots010\ldots01$ (n 1's, and each block of 0's has length $1 + |n|$). Note that a' and b exist in M by the $\mathbf{\Delta}_1^b$ bit–comprehension rule. It is straightforward that the counting array for X can be extracted from the product $a' \cdot b$.

The construction in (b) is done by reversing the above construction. Suppose that $\mathcal{N}$ is a model of $\mathbf{VTC}^0$. We can view $\mathcal{N}$ as a model of $\overline{\mathbf{VTC}}^0$, where the symbols of $\mathcal{L}_{\mathbf{FTC}^0}$ are interpreted according to their defining axioms given in Definition 3.10. We construct a model $\mathcal{M}$ for $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ by interpreting each second sort object X of $\mathcal{N}$ as the number

$$X(0)2^0 + \ldots + X(n-1)2^{n-1}, \tag{41}$$

(i.e., the number whose binary representation is $X(n-1)\ldots X(0)$) where $n = |X|$. All symbols of $\mathcal{L}_{\mathbf{\Delta}_1^b\text{-}\mathbf{CR}}$ except for $\cdot$ are interpreted in a straightforward manner. For example, if a is the number in $\mathcal{M}$ with the value from (41), for some second sort object $X \in \mathcal{N}$, then $|a| = |X|$ (more precisely, there is a second sort object $Z \in \mathcal{N}$ such that in $\mathcal{N}$, $|X| = Z(0)2^0 + \ldots + Z(m-1)2^{m-1}$, where $m = |Z|$, and $|a|$ is the number associated with Z). The axioms of $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ describing these symbols hold in $\mathcal{M}$ because their translations (except those involving $\cdot$) are easy theorems of $\overline{\mathbf{VTC}}^0$. (We will present a proof of the associativity of string addition in Appendix A.) It remains to (i) interpret $\cdot$, and prove its properties in $\mathcal{M}$, and (ii) show that other axioms of $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ are satisfied in $\mathcal{M}$. For (ii), the axiom scheme **Open-LIND** holds in $\mathcal{M}$, since **Open-IND** holds in $\overline{\mathbf{VTC}}^0$. Therefore we will present only the proof that the $\mathbf{\Delta}_1^b$ bit–comprehension rule is satisfied in $\mathcal{M}$.

4.3 Interpreting Multiplication for the Second Sort Objects in $\overline{\mathbf{VTC}}^0$

Now we need to define the "string multiplication" function $X \cdot_2 Y$ (we will simply write $X \cdot Y$), which is the binary representation of the product of the two numbers corresponding to X and Y by the mapping (41). It is known that the complexity of computing $X \cdot Y$ is $\mathbf{AC}^0$ complete for $\mathbf{TC}^0$ [8]. Thus our task is to formalize in $\mathbf{VTC}^0$ a $\mathbf{TC}^0$ algorithm computing this product. This can be reduced to computing the sum of n strings. The "school algorithm" is to write down the strings and sum up the bits in the same columns, starting from the lowest order bits, with carries from the previous columns. However, it might not be possible to formalize this algorithm in $\overline{\mathbf{VTC}}^0$, and we will formalize the algorithm from [3], where it is shown that multiplication can be defined in $\mathbf{TC}^0$-**Frege**.

4.3.1 Adding n Strings

Suppose that we are to add n strings, each of length $\leq m$ (written as a table of n rows and m columns). The methods from [3] is to divide the m columns into $2k$ blocks, each consisting of ℓ columns. (Thus each block has n substrings of length ℓ.) The numbers k and ℓ are chosen so that the sums of the substrings in the $2k$ blocks can be computed concurrently. It remains to use these subsums to obtain the desired sum; a further requirement for k and ℓ is that this last step can be carried out efficiently.

More precisely let $\ell = 1 + \lceil \log n \rceil$ and $k = \lceil m/2\ell \rceil$. Notice that the sum of the n substrings in each block is a string of length bounded by 2ℓ, or equivalently a number (i.e., first sort object of $\mathcal{N}$) which is $\leq 2^{2\ell} \leq 4n^2$. Let $b_0, \ldots, b_{2k-1}$ be the subsums. The sum of the original n strings is computed from $2k$ such "short" strings by first "concatenating" $b_0, b_2, \ldots, b_{2k-2}$ and "concatenating" $b_1, b_3, \ldots, b_{2k-1}$ (i.e., concatenating the binary string representations of $b_0, b_2, \ldots, b_{2k-2}$, and concatenating the binary string representations of $b_1, b_3, \ldots, b_{2k-1}$), then adding the 2 resulting strings together.

Formally, suppose that the n strings are represented as n rows $Z^{[0]}, \ldots, Z^{[n-1]}$ in an array Z (using the pairing function). Our goal is to compute their sum $\mathit{Sum}(n, m, Z)$ as a string function of n, m, Z.[6] Note that if for each i, $0 \leq i < m$, let c_i is the total number of bits in the ith column of Z, then

$$\mathit{Sum}(n, m, Z) = \sum_{i=0}^{m-1} 2^i c_i.$$

We will "store" c_i's in a string W, and then define $\mathit{Sum}(n, m, Z)$ as a function $\mathit{Sum}'(m, n, W)$.[7] Here W is a $\mathbf{TC}^0$ string function of n, m and Z: it has m rows, and the row $W^{[i]}$ of W has length exactly c_i. It can be defined as follows. First, let $\bar{Z}$ be the transpose of Z: for $0 \leq i < m$,

$$|\bar{Z}^{[i]}| \leq n \wedge \forall j < n \ \bar{Z}^{[i]}(j) \leftrightarrow Z(j, i).$$

Then the total number of bits in the ith column of Z is exactly $\mathit{numones}(n, \bar{Z}^{[i]})$, the total number of bits in the ith row of $\bar{Z}$. Now $W = \mathit{AddCols}(n, m, Z)$ where $\mathit{AddCols}(n, m, Z)$ is defined by

$$|\mathit{AddCols}(n, m, Z)| \leq \langle m, n \rangle \wedge \forall i < m, j < n \ \mathit{AddCols}(n, m, Z)(i, j) \leftrightarrow j < \mathit{numones}(n, \bar{Z}^{[i]}). \tag{42}$$

We need to compute

$$\mathit{Sum}'(m, n, W) = \sum_{i=0}^{m-1} 2^i |W^{[i]}|, \tag{43}$$

where n is a bound for $|W^{[0]}|, \ldots, |W^{[m-1]}|$: $|W^{[i]}| \leq n$ for $0 \leq i < m$.

Notice that the number functions 2^x where $x < |a|$ for some number $a \in \mathcal{N}$, and $\log x$, are in $\mathbf{FTC}^0$ (in fact, they are in $\mathbf{FAC}^0$, see [10] for defining the 2^x, for small x, in $\mathbf{V}^0$). Let ℓ and k be as in the above discussion, i.e.,

$$\ell = 1 + \lceil \log n \rceil, \qquad k = \lceil m/2\ell \rceil.$$

Write c_i for $|W^{[i]}|$, for $0 \leq i < m$. Divide $c_{m-1}, \ldots, c_0$ into $2k$ blocks of length ℓ each:

$$c_{2k\ell-1}, \ldots, c_{(2k-1)\ell}; \qquad \ldots; \qquad c_{2\ell-1}, \ldots, c_\ell; \qquad c_{\ell-1}, \ldots, c_0.$$

For $0 \leq i < 2k$, we will define b_i to be the sum of the ith block, $b_i = \sum_{j=0}^{\ell-1} 2^j c_{i\ell+j}$. Formally, this is a number function of W, i and ℓ, i.e., $b_i = \mathit{sum}(W, i\ell, \ell)$ where

$$\mathit{sum}(W, a, \ell) = \sum_{j=0}^{\ell-1} 2^j c_{a+j}$$

[6] Here n, m indicate the "size" of Z, i.e., it has n rows, each of length $\leq m$.
[7] Note the difference in ordering of n and m as arguments in Sum and Sum'.

(the sum of the block of length ℓ, starting from a). Here, *sum* can be defined using *numones*: $sum(W, a, \ell)$ is the number of bits in the "long" string Y,

$$sum(W, a, \ell) = numones(|Y|, Y),$$

where Y consists of 1 substring of c_a 1's; 2^1 substrings, each of c_{a+1} 1's; ...; $2^{\ell-1}$ substrings, each of $c_{a+\ell-1}$ 1's. Obviously, we can define such Y as an $\mathbf{AC}^0$ function of W:

$$|Y| \leq 2^\ell n \wedge \forall j < \ell \forall u < 2^j \forall v < n \; [Y((2^j - 1)n + un + v) \leftrightarrow v < c_{a+j}],$$

(note that n is a bound for $c_a, \ldots, c_{a+\ell-1}$). Also, $\overline{\mathbf{VTC}}^0$ proves the following properties of *sum*:

$$sum(W, a, 0) = c_a, \qquad sum(W, a, \ell + 1) = sum(W, a, \ell) + 2^\ell c_{a+\ell}, \qquad sum(W, a, \ell) < n2^\ell.$$

In particular, we have $b_i < 2^{2\ell}$, for $i < 2k$.

Let

$$L = \sum_{i=0}^{k-1} 2^{2i\ell} b_{2i}, \qquad H = \sum_{i=0}^{k-1} 2^{(2i+1)\ell} b_{2i+1}.$$

Since $b_i < 2^{2\ell}$ for $i < 2k$, L and H can be computed simply by concatenating the binary representations of $b_0, b_2, \ldots, b_{2k-2}$ and $b_1, b_3, \ldots, b_{2k-1}$, respectively. (More precisely, we may have to pad each b_i with leading 0's to make them of length exactly 2ℓ, and then concatenate these strings of equal length.)

Now

$$Sum'(m, n, W) = \sum_{i=0}^{2k-1} 2^{i\ell} b_i = L + H.$$

As a result, $Sum(n, m, Z) = Sum'(m, n, W)$ is a function of $\mathcal{L}_{\mathbf{FTC}^0}$. Thus we can define $X \cdot Y$ as follows.

Given X and Y, let $X \otimes Y$ be the "table" that we use in the "school algorithm" to multiply X and Y, $X \otimes Y = Z$ where

$$|Z| \leq (|X| + |Y|)|Y| \wedge \forall x < |X| \forall y < |Y|, \; Z(y, x + y) \leftrightarrow [X(x) \wedge Y(y)]. \tag{44}$$

(Z has $|Y|$ rows, each is of length $\leq |X| + |Y|$.) Then $X \cdot Y = Sum(|Y|, |X| + |Y|, Z)$. It follows that $X \cdot Y$ is a function of $\mathcal{L}_{\mathbf{FTC}^0}$. It remains to prove the properties of this function, i.e., it is commutative, and distributive over $X + Y$.

4.3.2 Proving Properties Of $X \cdot Y$

First we need to show that $\cdot_2$ is commutative.

Lemma 4.1. $\overline{\mathbf{VTC}}^0 \vdash X \cdot Y = Y \cdot X$.

Proof. Recall that we define $Sum(n, m, Z) = Sum'(m, n, W)$, where $W = AddCols(n, m, Z)$ which is defined in (42). Thus it suffices to show that

$$AddCols(|Y|, m, X \otimes Y) = AddCols(|X|, m, Y \otimes X), \tag{45}$$

where $m = |X| + |Y|$.

Let $Z_1 = X \otimes Y$ and $Z_2 = Y \otimes X$. Notice that for $i < m$ the column $\bar{Z}_1^{[i]}$ of Z_1 and column $\bar{Z}_2^{[i]}$ of Z_2 are just permutation of each other. In particular, $|\bar{Z}_1^{[i]}|, |\bar{Z}_2^{[i]}| \leq i + 1$, and

$$\bar{Z}_1^{[i]}(y) \leftrightarrow y \leq i \wedge Y(y) \wedge X(i - y), \qquad \bar{Z}_2^{[i]}(x) \leftrightarrow x \leq i \wedge X(x) \wedge Y(i - x),$$

and hence $\bar{Z}_1^{[i]}(y) \leftrightarrow \bar{Z}_2^{[i]}(i-y)$, for $y \le i$. To prove (45), we will show that for $i < m$, $\bar{Z}_1^{[i]}$ and $\bar{Z}_2^{[i]}$ have the same number of elements, i.e.,

$$numones(i, \bar{Z}_1^{[i]}) = numones(i, \bar{Z}_2^{[i]}).$$

It suffices to prove more generally that if there is an one-one mapping between $\bar{Z}_1[i]$ and $\bar{Z}_2[i]$, then they have the same number of elements. This is proved in the next lemma. □

In the following lemma, suppose that there is an one–one mapping (specified by M) between the initial segments $\{i \mid i \in X \wedge i < \ell\}$ and $\{j \mid j \in Y \wedge j < \ell\}$ of X and Y respectively. Then these initial segments have the same number of elements.

Lemma 4.2. *Let ℓ, X, Y, M be such that*

$$\forall i < \ell \exists! j < \ell M(i,j) \ \wedge \ \forall j < \ell \exists! i < \ell M(i,j), \qquad \textit{and} \qquad \forall i < \ell,\ X(i) \leftrightarrow \exists j < \ell (M(i,j) \wedge Y(j)). \quad (46)$$

Then, $\mathbf{VTC}^0 \vdash numones(\ell, X) = numones(\ell, Y)$.

Proof. First, from (46) it is easy to see that

$$\forall j < \ell,\ Y(j) \leftrightarrow \exists i < \ell (X(i) \wedge M(i,j)).$$

Let Z be the string such that $Z^{[k]}$ is the image of the initial segment $\{i \mid i \in X \wedge i < k\}$ of X, i.e.,

$$\forall k < \ell \forall j < \ell\ [Z^{[k]}(j) \leftrightarrow \exists i < k (M(i,j) \wedge X(i))].$$

Then, $Z^{[\ell]} = Y$. We can prove by induction on k that $numones(k, X) = numones(\ell, Z^{[k]})$. Consequently, $numones(\ell, X) = numones(\ell, Y)$. □

Now we will show that $\cdot_2$ is distributive over $+_2$.

Lemma 4.3. $\overline{\mathbf{VTC}}^0 \vdash X \cdot (Y + Z) = X \cdot Y + X \cdot Z$.

Proof. It suffices to prove

$$X^{<i} \cdot (Y + Z) = X^{<i} \cdot Y + X^{<i} \cdot Z \tag{47}$$

by induction on i, where $X^{<i}$ is the i low-order bits of X; that is

$$X^{<i} = \{j < i \mid X(j)\}.$$

The base case follows from the fact that $\mathbf{0} \cdot Y = \mathbf{0}$, which can be proved from the definition of $X \cdot Y$.

For the induction step there are two cases: $X^{<i+1} = X^{<i}$ and $X^{i+1} = X^{<i} + \{i\}$. Since the first case is trivial, we consider the second case. To simplify notation, we will write X for $X^{<i}$ and write X' for $X + \{i\} = X^{<i+1}$.

Thus our task is to prove in $\mathbf{VTC}^0$

$$X' \cdot (Y + Z) = X' \cdot Y + X' \cdot Z \tag{48}$$

from the induction hypothesis

$$|X| \le i \ \wedge \ X \cdot (Y + Z) = X \cdot Y + X \cdot Z.$$

We need the following fact, which we prove below:

$$\overline{\mathbf{VTC}}^0 \vdash |X| \le i \supset (X + \{i\}) \cdot Y = X \cdot Y + \{i\} \cdot Y. \tag{49}$$

From the definition of $X \cdot Y$ we have

$$X \cdot \{i\} = \{x + i \mid x \in X\} \tag{50}$$

From the commutativity of $\cdot_2$ (Lemma 4.1), the associativity of $+_2$ (Lemma A.2), and (50) we can derive that

$$\{i\} \cdot (Y + Z) = (Y + Z) \cdot \{i\} = Y \cdot \{i\} + Z \cdot \{i\} = \{i\} \cdot Y + \{i\} \cdot Z. \tag{51}$$

Now we prove (48) as follows, using associativity and commutativity of $+_2$.

$$\begin{aligned} X' \cdot (Y + Z) &= X \cdot (Y + Z) + \{i\} \cdot (Y + Z) \qquad \text{[by (49)]} \\ &= X \cdot Y + X \cdot Z + \{i\} \cdot (Y + Z) \qquad \text{[by Ind Hyp]} \\ &= X \cdot Y + X \cdot Z + \{i\} \cdot Y + \{i\} \cdot Z \qquad \text{[by (51)]} \\ &= X' \cdot Y + X' \cdot Z \qquad \text{[by (49)].} \end{aligned}$$

It remains to prove (49). Using the commutativity of $\cdot_2$, rewrite the equality in (49) as

$$Y \cdot (X + \{i\}) = Y \cdot X + Y \cdot \{i\}.$$

To prove this, it suffices to prove that

$$\mathit{Sum}(i, i + |Y|, Y \otimes (X + \{i\})) = \mathit{Sum}(|X|, |X| + |Y|, Y \otimes X) + Y \cdot \{i\}.$$

Notice that since $|X| \leq i$, the "table" $Z_1 = Y \otimes (X + \{i\})$ is exactly $Y \otimes X$ appended with an additional row

$$Z_1^{[i]} = \{y + i \mid y \in Y\} = Y \cdot \{i\}.$$

Therefore (49) follows from the next lemma. □

Lemma 4.4. *Suppose that* $|Z^{[i]}| \leq m$, *for* $0 \leq i \leq n$. *Then* $\overline{\mathbf{VTC}}^0 \vdash \mathit{Sum}(n+1, m, Z) = \mathit{Sum}(n, m, Z) + Z^{[n]}$.

Proof. We have defined Sum using Sum'. (Recall the definition of Sum' in (43).) We need to show that

$$\mathit{Sum}'(m, n + 1, W) = \mathit{Sum}'(m, n, W_1) + Z^{[n]}$$

where

$$W_1 = \mathit{AddCols}(n, m, Z), \qquad W = \mathit{AddCols}(n + 1, m, Z),$$

It is straightforward that for $i < m$,

$$|W^{[i]}| = \begin{cases} |W_1^{[i]}| + 1 & \text{if } i \in Z^{[n]} \\ |W_1^{[i]}| & \text{if } i \notin Z^{[n]}. \end{cases} \tag{53}$$

We will prove by induction on $m' \leq m$ that

$$\mathit{Sum}'(m', n + 1, W) = \mathit{Sum}'(m', n, W_1) + (Z^{[n]})^{<m'}, \tag{54}$$

where $X^{<z}$ is the initial segment $\{x \mid x \in X \wedge x < z\}$ of X.

For the base case, (54) obviously holds when $m' = 0$.

For the induction step, suppose that (54) holds for some $m' < m$. Note that by the definition of Sum' (43), intuitively,

$$\mathit{Sum}'(m' + 1, n + 1, W) = \mathit{Sum}'(m', n + 1, W) + 2^{m'}|W^{[m']}|.$$

Formally, let $\mathit{ToString}(c, \ell)$ be the set

$$\mathit{ToString}(c, \ell) = \{i + \ell \mid \mathit{BIT}(i, c)\}.$$

Then we can show from the definition of Sum' that

$$Sum'(m'+1, n+1, W) = Sum'(m', n+1, W) + ToString(c_{m'}, m'),$$

(where $c_{m'}$ stands for $|W^{[m']}|$). Similarly,

$$Sum'(m'+1, n, W_1) = Sum'(m', n, W_1) + ToString(c^1_{m'}, m'),$$

where $c^1_{m'}$ stands for $|W_1^{[m']}|$.

By the induction hypothesis, it remains to show that

$$ToString(c_{m'}, m') + (Z^{[n]})^{<m'} = ToString(c^1_{m'}, m') + (Z^{[n]})^{<m'+1}.$$

This follows from (53) and the definition of *ToString*. □

4.4 Interpreting the $\mathbf{\Delta}_1^b$ Comprehension Rule in $\mathbf{VTC}^0$

Recall the definition of the $\mathbf{\Delta}_1^b$ bit–comprehension rule (in single–sorted logic) in section 4.1. Note that this rule specifies an inductive definition of $\mathbf{\Delta}_1^b$-**CR**. In order to show that $\mathcal{M}$ is a model of $\mathbf{\Delta}_1^b$-**CR**, where $\mathcal{M}$ is the structure constructed from a model $\mathcal{N}$ of $\mathbf{VTC}^0$ as discussed in (b) of section 4.2.1, we will show that $\mathbf{VTC}^0$ (and $\overline{\mathbf{VTC}}^0$) satisfies the $\mathbf{g\Delta}_1^B$ *comprehension rule*, the two–sorted version of the $\mathbf{\Delta}_1^b$ bit–comprehension rule. This rule for two–sorted theories is obtained from the $\mathbf{\Delta}_1^b$ bit–comprehension rule using the syntactic translation $\psi \rightharpoonup \psi^\sharp$ described in section 4.2.1.

Recall the definition of $\mathbf{g\Sigma}_1^B$ and $\mathbf{g\Pi}_1^B$ formulas given in section 2.1. Note that under the translation $\psi \rightharpoonup \psi^\sharp$, $\mathbf{\Sigma}_1^b$ and $\mathbf{\Pi}_1^b$ formulas translate to formulas equivalent to those in $\mathbf{g\Sigma}_1^B$ and $\mathbf{g\Pi}_1^B$, respectively.

The $\mathbf{g\Delta}_1^B$ comprehension rule is defined as follows.

Definition 4.5. *A (two–sorted) theory $\mathcal{T}$ over $\mathcal{L}$ is said to admit the $\mathbf{g\Delta}_1^B(\mathcal{L})$ comprehension rule if whenever*

$$\mathcal{T} \vdash \forall z < b,\ \varphi(z) \leftrightarrow \psi(z), \qquad \textit{then} \qquad \mathcal{T} \vdash \exists X \le b \forall z < b,\ X(z) \leftrightarrow \varphi(z)$$

where φ is a $\mathbf{g\Sigma}_1^B(\mathcal{L})$ formula, and ψ is a $\mathbf{g\Pi}_1^B(\mathcal{L})$ formula (φ and ψ may have other free variables).

We omit $\mathcal{L}$ from $\mathbf{g\Delta}_1^B(\mathcal{L})$ when it is clear from context.

This rule is apparently weaker than the $\mathbf{g\Delta}_1^B$ comprehension axiom. In particular, $\mathbf{VTC}^0$ may not prove the axiom.

Our task for this section to prove the following theorem.

Theorem 4.6. $\mathbf{VTC}^0$ *and* $\overline{\mathbf{VTC}}^0$ *admit the* $\mathbf{g\Delta}_1^B$ *comprehension rule.*

Note that it suffices to prove the theorem for $\overline{\mathbf{VTC}}^0$, since this theory is conservative over $\mathbf{VTC}^0$. We will use the $\mathbf{g\Sigma}_1^B$ replacement rule, defined as follows.

Definition 4.7. *A (two–sorted) theory $\mathcal{T}$ over $\mathcal{L}$ is said to admit the $\mathbf{g\Sigma}_1^B(\mathcal{L})$ replacement rule if whenever*

$$\mathcal{T} \vdash \forall z < b \exists Z < b \varphi(z, Z), \qquad \textit{then} \qquad \mathcal{T} \vdash \exists W < \langle b, b\rangle \forall z < b\ \varphi(z, W^{[z]}),$$

for any $\mathbf{g\Sigma}_1^B(\mathcal{L})$ formula φ which may contain other free variables.

Note that if $\mathcal{T}$ admits the $\mathbf{g\Sigma}_1^B$ replacement rule, then each $\mathbf{g\Sigma}_1^B$ theorem of $\mathcal{T}$ is provably equivalent in $\mathcal{T}$ to a $\mathbf{\Sigma}_1^B$ formula.

Lemma 4.8. *If the theory $\mathcal{T}$ (extending $\mathbf{V}^0(Row)$) proves $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** and admits the $\mathbf{g\Sigma}_1^B(\mathcal{L})$ replacement rule, then it also admits the $\mathbf{g\Delta}_1^B(\mathcal{L})$ comprehension rule.*

Proof. Suppose that
$$\mathcal{T} \vdash \forall z < b,\ \varphi(z) \leftrightarrow \psi(z),$$
where φ is a $\mathbf{g\Sigma}_1^B(\mathcal{L})$ formula, and ψ is a $g\mathbf{\Pi}_1^B(\mathcal{L})$ formula which may have other free variables. Then
$$\mathcal{T} \vdash \forall z < b,\ \varphi(z) \vee \neg\psi(z).$$

Therefore
$$\mathcal{T} \vdash \forall z < b \exists Z \leq 1,\ [Z(0) \wedge \varphi(z)] \vee [\neg Z(0) \wedge \neg\psi(z)].$$
Now $\theta(z, Z) \equiv [Z(0) \wedge \varphi(z)] \vee [\neg Z(0) \wedge \neg\psi(z)]$ is equivalent to a $\mathbf{g\Sigma}_1^B$ formula. Since $\mathcal{T}$ admits the $\mathbf{g\Sigma}_1^B(\mathcal{L})$ replacement rule,
$$\mathcal{T} \vdash \exists W < \langle b, b\rangle \forall z < b\ \theta(z, W^{[z]}).$$

Let X be defined by $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP**:
$$|X| \leq b \wedge \forall z < b,\ X(z) \leftrightarrow W^{[z]}(0).$$
Then obviously $\forall z < b,\ X(z) \leftrightarrow \varphi(z)$. □

Now Theorem 4.6 follows from the following lemma.

Lemma 4.9. $\overline{\mathbf{VTC}}^0$ *admits the $\mathbf{g\Sigma}_1^B(\mathcal{L}_{\mathbf{FTC}^0})$ replacement rule.*

Proof. Suppose that
$$\overline{\mathbf{VTC}}^0 \vdash \forall z < b \exists Z < b \varphi(z, Z),$$
for some $\mathbf{g\Sigma}_1^B(\mathcal{L}_{\mathbf{FTC}^0})$ formula $\varphi(z, Z)$. By the $\mathbf{g\Sigma}_1^B$ Witnessing Theorem (Theorem 3.20) there is a function $F(z)$ of $\mathcal{L}_{\mathbf{FTC}^0}$ such that
$$\overline{\mathbf{VTC}}^0 \vdash \forall z < b \varphi(z, F(z)).$$
Let G be defined as follows:
$$|G| \leq \langle b, b\rangle \wedge \forall z < b G^{[z]} = F(z).$$
Then we have
$$\forall z < b \varphi(z, G^{[z]}).$$
Also, G is a function of $\mathcal{L}_{\mathbf{FTC}^0}$. Therefore $\overline{\mathbf{VTC}}^0 \vdash \exists W \leq \langle b, b\rangle \forall z < b \varphi(z, W^{[z]})$. □

5 Conclusion

We show (Theorem 2.6) that the $\mathbf{TC}^0$ relations are precisely those represented by $\mathbf{\Sigma}_0^{B,Th}$ formulas. We also present the finitely axiomatizable, second–order theory $\mathbf{VTC}^0$ which characterizes $\mathbf{TC}^0$ in the same way that Buss's theories $\mathbf{S}_2^i$'s characterize the polynomial time hierarchy. Our characterization of $\mathbf{TC}^0$ by the theory $\mathbf{VTC}^0$ is based on the fact that counting the number of 1 bits in a string is complete for $\mathbf{TC}^0$ rather than the "hidden power" of the multiplication function (which is also complete for $\mathbf{TC}^0$) usually present *a prior* in first–order theories.

We show that a number of combinatorial problems are provable in $\mathbf{VTC}^0$. In particular, we show that $\mathbf{VTC}^0$ is RSUV isomorphic to Johannsen and Pollett's "minimal" theory $\mathbf{\Delta}_1^b$-**CR**. The main part of proving this RSUV isomorphism is in defining (string) multiplication and proving its properties. The RSUV isomorphism between $\mathbf{VTC}^0$ and $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ shows that $\mathbf{\Delta}_1^b$-$\mathbf{CR} = \mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ for some constant i, answering a question by [21].

In addition, we show that a form of the Pigeonhole Principle is provable in $\mathbf{VTC}^0$. In [6] Buss shows that the STCONN tautologies (and thus the HEX tautologies) have polynomial size constant depth $\mathbf{TC}^0$–**Frege** proofs. It can be seen that his arguments can be formalized in our theory $\mathbf{VTC}^0$. The proofs of these principles in $\mathbf{VTC}^0$ are more uniform that the $\mathbf{TC}^0$–**Frege** proofs.

In [16], it is shown that division is in uniform $\mathbf{TC}^0$. Thus the (string) division function might be $\mathbf{\Sigma}_1^B$–definable in $\mathbf{VTC}^0$. An interesting problem is to formalize the algorithm of [16] in $\mathbf{VTC}^0$.[8]

We are also able to generalize the method used in developing $\mathbf{VTC}^0$ to obtain a scheme of theories characterizing a number of other subclasses of $\mathbf{P}$. Our work follows the program outlined in [10, 11], which proposes defining and studying second-order theories and propositional proof systems associated with various complexity subclasses of $\mathbf{P}$. We have not treated the connection with propositional proof systems here, but this is the subject of ongoing investigation. It is also of interest to compare various theories, e.g., **VNC** vs. $\mathbf{U}_1^1$ [11].

A Interpreting Addition for the Second Sort Objects in $\overline{\mathbf{V}}^0$

We define the "string addition" function $X +_2 Y$ (we will simply write $X + Y$, the meaning will be clear from the context), which is the binary representation of the sum of the two numbers corresponding to X and Y by the mapping (41).

We will show that the string function $X + Y$ can be defined in any model of $\mathbf{V}^0$. Here addition is defined using the conventional algorithm, i.e., adding digits of the same order in X and Y together with carries from the previous result. More precisely, let $\varphi_+(i, X, Y)$ represent the carry at the bit position i when adding X and Y. Then the ith bit of the sum $X + Y$ is

$$(X + Y)(i) \Leftrightarrow X(i) \oplus Y(i) \oplus \varphi_+(i, X, Y)$$

(Here $\oplus$ stands for *exclusive or*, i.e., $p \oplus q \leftrightarrow (p \wedge \neg q) \vee (\neg p \wedge q)$.)

Definition A.1. *Let $\varphi_+(i, X, Y)$ be the $\mathbf{\Sigma}_0^B$ formula*

$$\exists j < i,\ X(j) \wedge Y(j) \wedge \forall \ell < i[j < \ell \supset X(\ell) \oplus Y(\ell)]. \tag{57}$$

Then $X + Y$ is defined as follows:

$$|X + Y| \leq |X| + |Y| \ \wedge\ \forall i < |X| + |Y|,\ (X + Y)(i) \leftrightarrow [X(i) \oplus Y(i) \oplus \varphi_+(i, X, Y)].$$

Since the above definition is symmetric for X and Y, it follows that $\mathbf{V}^0 \vdash X + Y = Y + X$. It remains to show the associativity for this function.

Lemma A.2. $\overline{\mathbf{V}}^0 \vdash X + (Y + Z) = (X + Y) + Z$.

Proof. It suffices to show that for $i < |X| + |Y| + |Z|$,

$$(X + (Y + Z))(i) \leftrightarrow ((X + Y) + Z)(i).$$

This is equivalent to

$$X(i) \oplus (Y + Z)(i) \oplus \varphi_+(i, X, Y + Z) \leftrightarrow (X + Y)(i) \oplus Z(i) \oplus \varphi_+(i, X + Y, Z).$$

[8] This problem was suggested by Albert Atserias.

From (57), the above is simplified to

$$\varphi_+(i, Y, Z) \oplus \varphi_+(i, X, Y + Z) \leftrightarrow \varphi_+(i, X, Y) \oplus \varphi_+(i, X + Y, Z).$$

Let a_i, b_i, c_i and d_i denote $\varphi_+(i, Y, Z)$, $\varphi_+(i, X, Y + Z)$, $\varphi_+(i, X, Y)$ and $\varphi_+(i, X + Y, Z)$ respectively. We need to show that

$$a_i \oplus b_i \leftrightarrow c_i \oplus d_i,$$

for $i < |X| + |Y| + |Z|$. We will prove a stronger result, i.e., (a_i, b_i) is a permutation of (c_i, d_i). In particular, we will show by induction on i that

$$(a_i \wedge b_i \leftrightarrow c_i \wedge d_i) \wedge (a_i \vee b_i \leftrightarrow c_i \vee d_i). \tag{58}$$

The base case is trivial, since

$$\mathbf{V}^0 \vdash \neg a_0 \wedge \neg b_0 \wedge \neg c_0 \wedge \neg d_0.$$

The induction step follows from the inductive evaluation of a_i b_i, c_i and d_i:

$$\begin{aligned} a_{i+1} &= [Y(i) \wedge Z(i)] \vee [(Y(i) \oplus Z(i)) \wedge a_i], \\ b_{i+1} &= [X(i) \wedge (Y(i) \oplus Z(i) \oplus a_i)] \vee [(X(i) \oplus Y(i) \oplus Z(i) \oplus a_i) \wedge b_i], \\ c_{i+1} &= [X(i) \wedge Y(i)] \vee [(X(i) \oplus Y(i)) \wedge c_i], \\ d_{i+1} &= [Z(i) \wedge (X(i) \oplus Y(i) \oplus c_i)] \vee [(X(i) \oplus Y(i) \oplus Z(i) \oplus c_i) \wedge d_i]. \end{aligned}$$

It remains to verify (58) for $i + 1$. This can be done by using the induction hypothesis and the above properties, together with checking all possible values of $X(i), Y(i)$ and $Z(i)$. Details are omitted. $\Box$

B Proving the Pigeonhole Principle in $\mathbf{VTC}^0$

We give an example of reasoning in $\mathbf{VTC}^0$ by formalizing and proving the Pigeonhole Principle (*PHP*) in $\mathbf{VTC}^0$. This principle states that for any mapping from a set of a numbers to a set of $(a-1)$ numbers, there must be 2 numbers in the domain that have the same image. We will formalize and prove this principle in $\mathbf{VTC}^0$. In the following definition, the mapping is represented by the set X of pairs of pre-images and images ($Y(y, z)$ holds if y is the image of z).

Theorem B.1.

$$\mathbf{VTC}^0 \vdash \forall z \le a\ \exists y < a\ X(y, z) \supset \exists y < a\ \exists z_1 \le a\ \exists z_2 < z_1,\ X(y, z_1) \wedge X(y, z_2). \tag{59}$$

Proving *PHP* involves formalizing a number of concepts, such as set union, total number of bits in an array, etc. We will define these functions below, and it is straightforward that they are members of $\mathcal{L}_{\mathbf{FTC}^0}$.

Union: $\mathit{Union}(b, X, Y)(z) \leftrightarrow z < b \wedge (X(z) \vee Y(z))$.

We interpret Z as an array of a rows, and each row has length bounded by b.

Finite union: $\mathit{FiniteUnion}(a, b, Z)(z) \leftrightarrow z < b \wedge \exists y < aZ^{[y]}(z)$.

Total number of bits in an array: $\mathit{totNumones}(a, b, Z) = \mathit{numones}(ab, F_0(a, b, Z))$, where $F_0(a, b, Z)$ is the function of Z which concatenates all the rows of the array Z:

$$F_0(a, b, Z)(ax + y) \leftrightarrow Z^{[x]}(y),\ \text{for } x < a,\ y < b$$

Lemma B.2. *The following are theorems of* $\overline{\mathbf{VTC}}^0$*:*
a) $numones(b, Union(b, X, Y)) \leq numones(b, X) + numones(b, Y)$.
b) $totNumones(a + 1, b, Z) = totNumones(a, b, Z) + numones(b, Z^{[a]})$.
c) $numones(b, FiniteUnion(a, b, Z)) \leq totNumones(a, b, Z)$.
d) $\forall x < a \; numones(b, Z^{[x]}) \leq k \;\supset\; totNumones(a, b, Z) \leq ak$.

Proof. Part a) is proved by induction on b. Part b) is proved by noting that $F_0(a+1, b, Z)$ is the concatenation of $F_0(a, b, Z)$ and $Z^{[a]}$.

For c), the proof is by induction on a. The base case is straightforward. For the induction step, note that

$$FiniteUnion(a + 1, b, Z) = Union(b, FiniteUnion(a, b, Z), Z^{[a]})$$

We have

$$\begin{aligned}
& numones(b, FiniteUnion(a + 1, b, Z)) \\
= \; & numones(b, Union(b, FiniteUnion(a, b, Z), Z^{[a]})) \\
\leq \; & numones(b, FiniteUnion(a, b, Z)) + numones(b, Z^{[a]}) \quad \text{(by a.)} \\
\leq \; & totNumones(a, b, Z) + numones(b, Z^{[a]}) \quad \text{(by the I.H.)} \\
= \; & totNumones(a + 1, b, Z) \quad \text{(by b.)}
\end{aligned}$$

Finally, part d) is proved by induction on a, using part b). □

Proof of Theorem B.1. We have to show that there exists a row of X that contains at least 2 bits. We will prove by contradiction, by showing that if every row of X has at most 1 bit, then the total number of bits in the array X is at most a. On the other hand, the union of the rows of X has $(a + 1)$ bits. This contradicts part c) of Lemma B.2. Details are as follows.

Suppose that $\forall y < a \; numones(a + 1, X^{[y]}) \leq 1$. By Lemma B.2 part d), $totNumones(a, a + 1, Z) \leq a$. By Lemma B.2 part c), $numones(a + 1, FiniteUnion(a, a + 1, Z)) \leq a$. However, it is obvious that $\forall z \leq a \; FiniteUnion(a, a+1, Z)(z)$. By a simple induction argument, this implies $numones(a+1, FiniteUnion(a, a+1, Z)) = a + 1$, a contradiction. □

References

[1] Toshiyasu Arai. Bounded arithmetic AID for Frege system. *Annals of Pure and Applied Logic*, 103:155–199, 2000.

[2] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On Uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[3] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On Interpolation and Automatization for Frege Systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[4] Samuel Buss. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[5] Samuel Buss. The Boolean formula value problem is in ALOGTIME. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 123–131, 1987.

[6] Samuel Buss. Polynomial-size Frege and Resolution Proofs of st-Connectivity and Hex Tautologies. `http://math.ucsd.edu/~sbuss/ResearchWeb/`, 2003.

[7] Samuel R. Buss, editor. *Handbook of Proof Theory*. Elsevier, Amsterdam, 1998.

[8] Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant Depth Reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984.

[9] Peter Clote and Gaisi Takeuti. First Order Bounded Arithmetic and Small Boolean Circuit Complexity Classes. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II.* Birkhäuser, 1995.

[10] Stephen Cook. Proof Complexity and Bounded Arithmetic. Course Notes for CSC 2429S. `http://www.cs.toronto.edu/~sacook/`.

[11] Stephen Cook. Theories for Complexity Classes and Their Propositional Translations. To appear.

[12] Stephen Cook and Antonina Kolokolova. A Second-order System for Polytime Reasoning Based on Grädel's Theorem. *Annals of Pure and Applied Logic*, pages 193–231, 2003.

[13] Stephen Cook and Antonina Kolokolova. A Second-order Theory for NL. In *Logic in Computer Science (LICS)*, 2004.

[14] Stephen Cook and Tsuyoshi Morioka. Quantified Propositional Calculus and a Second-Order Theory for NC^1. Manuscript, submitted, 2004.

[15] Stephen Cook and Neil Thapen. The Strength of Replacement in Weak Arithmetic. In *Proc. 19th IEEE Symposium on Logic in Computer Science*, 2004.

[16] William Hesse. Division is in Uniform $\mathbf{TC}^0$. In *Eighth International Colloquium on Automata, Languages and Programming (ICALP 2001)*, 2001.

[17] Neil Immerman. *Descriptive Complexity.* Springer, 1999.

[18] Jan Johannsen. A Bounded Arithmetic Theory for Constant Depth Threshold Circuits. In Petr Hájek, editor, *GÖDEL '96. Springer Lecture Notes in Logic 6*, 1996.

[19] Jan Johannsen. Equational calculi and constant-depth propositional proofs. In Paul Beame and Samuel Buss, editors, *Proof Complexity and Feasible Arithmetics*, volume 39. AMS DIMACS Series, 1998.

[20] Jan Johannsen and Chris Pollett. On Proofs about Threshold Circuits and Counting Hierarchies. In *Proc. 13th IEEE Symposium on Logic in Computer Science*, pages 444–452, 1998.

[21] Jan Johannsen and Chris Pollett. On the Δ_1^b-Bit-Comprehension Rule. In Sam Buss, Petr Hájek and Pavel Pudlák, editor, *Logic Colloquium 98*, 2000.

[22] Antonina Kolokolova. *Systems of Bounded Arithmetic from Descriptive Complexity.* PhD thesis, University of Toronto, 2004.

[23] Jan Krajíček. Exponentiation and second-order bounded arithmetic. *Annals of Pure and Applied Logic*, 48:261–276, 1990.

[24] Jan Krajíček. On Frege and Extended Frege Proof Systems. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II.* Birkhäuser, 1995.

[25] Phuong Nguyen. VTC^0: A Second-Order Theory for TC^0. Master's thesis, University of Toronto, 2004. `http://www.cs.toronto.edu/~ntp/`.

[26] Phuong Nguyen and Stephen Cook. VTC^0: A Second-Order Theory for TC^0. In *Proc. 19th IEEE Symposium on Logic in Computer Science*, 2004.

[27] Rohit Parikh. Existence and feasibility in arithmetic. *Journal of Symbolic Logic*, 36(3):494–508, 1971.

[28] Alexander A. Razborov. An Equivalence between Second Order Bounded Domain Bounded Arithmetic and First Order Bounded Arithmetic. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 247–277. Oxford, 1993.

[29] Gaisi Takeuti. RSUV Isomorphism. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–386. Oxford, 1993.

[30] Domenico Zambella. Notes on Polynomially Bounded Arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.

[31] Domenico Zambella. End Extensions of Models of Linearly Bounded Arithmetic. *Annals of Pure and Applied Logic*, 88:263–277, 1997.