Theories for Complexity Classes and their Propositional Translations

Stephen Cook

April 17, 2004

Abstract

We present in a uniform manner simple two-sorted theories corresponding to each of eight complexity classes between $\mathbf{AC^0}$ and $\mathbf{P}$. We present simple translations between these theories and systems of the quantified propositional calculus.

1 Introduction

An important part of bounded arithmetic is associating logical theories with various complexity classes, and then translating proofs in these theories to families of proofs in appropriate propositional (or quantified propositional) proof systems. Our purpose here is to give a unified treatment of this three-way association, as it applies to the sequence of complexity classes

$$\mathbf{AC^0} \subset \mathbf{AC^0}(2) \subset \mathbf{TC^0} \subseteq \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC} \subseteq \mathbf{P} \tag{1}$$

where $\mathbf{P}$ is the class of problems solvable in polynomial time. Each of these classes has both a uniform and a nonuniform version. The idea is that proofs in the associated theory are restricted to using concepts in the uniform class, and lines of proofs in the associated propositional proof system are restricted to expressing concepts in the nonuniform class.

Consider, for example, the class $\mathbf{NC}^1$. The uniform version is **Alogtime**, the class of problems solvable by an alternating Turing machine in time $O(\log n)$. The associated theory **AID** [3] has predicates for **Alogtime** relations defined by log depth tree recursion, and allows induction on open formulas built from such predicates. A problem in nonuniform $\mathbf{NC}^1$ is defined by a polynomial-size family of log-depth Boolean circuits, or equivalently a polynomial-size family of propositional formulas. The corresponding propositional proof systems are called *Frege systems* and are described in standard logic textbooks: a Frege proof of a tautology A consists of a sequence of propositional formulas ending in A, where each formula is either an axiom or follows from earlier formulas by a rule of inference. Universal theorems of **AID** translate into polynomial size families of Frege proofs. More generally, bounded theorems of **AID** translate into polynomial size G_0 proofs, where G_0 [20] is a quantified generalization of a Frege system. Finally, **AID** proves the soundness of Frege systems, but not of any more powerful proof system.

In the literature there is a hodge-podge of theories associated with these complexity classes. Some theories are equational, some are single-sorted, some are two-sorted. Their underlying languages differ, even among single-sorted theories, and the manner in which they characterize complexity classes differs.

Here we associate a canonical minimal two-sorted finitely-axiomatizable theory with each of the above complexity classes. The association is depicted as follows.

$$\begin{array}{lllllllll} \text{class} & \mathbf{AC^0} & \mathbf{AC^0}(2) & \mathbf{TC^0} & \mathbf{NC}^1 & \mathbf{L} & \mathbf{NL} & \mathbf{NC} & \mathbf{P} \\ \text{theory} & \mathbf{V^0} & \mathbf{V^0}(2) & \mathbf{VTC}^0 & \mathbf{VNC}^1 & \mathbf{VL} & \mathbf{V\text{-}Krom} & \mathbf{U}^1 & \mathbf{TV}^0 \end{array} \tag{2}$$

None of these theories is exactly new except our theory $\mathbf{TV}^0$ for polynomial time, but most of the theories are not well-known.

Each of the complexity classes $\mathbf{C}$ has an associated class $\mathbf{FC}$ of functions consisting of those p-bounded functions whose bit graphs are in $\mathbf{C}$. In each case, the theory associated with the class has the property that the provably total functions in the theory (using Σ_1^1-formulas to describe the function graphs) are exactly those functions in the class $\mathbf{FC}$.

Each of the theories in (2) has the same "second-order" underlying language $\mathcal{L}_A^2$ [48, 49]. The language $\mathcal{L}_A^2$ is actually a language for the two-sorted first-order predicate calculus, where one sort is for numbers in $\mathbb{N}$ and the second sort is for finite sets of numbers. Here we regard an object of the second sort as a finite string over the alphabet $\{0,1\}$ (the i-th bit in the string is 1 iff i is in the set). The strings are the objects of interest for the complexity classes, and serve as the main inputs for the machines or circuits that determine the class. The numbers serve a useful purpose as indices for the strings when describing properties of the strings. When they are used as machine or circuit inputs, they are presented in unary notation.

In the more common single-sorted theories such as the hierarchies $\mathbf{S}_2^i$ and $\mathbf{T}_2^i$ [4] the underlying objects are numbers which are presented in binary notation as inputs to Turing machines. Our two-sorted treatment has the advantage that the underlying language has no primitive operations on strings except the length function $|X|$ and the bit predicate $X(i)$ (meaning $i \in X$). This is especially important for studying weak complexity classes such as $\mathbf{AC^0}$. The standard language for single-sorted theories includes number multiplication, which is not an $\mathbf{AC^0}$ function on binary strings. Even when studying classes such as $\mathbf{TC^0}$ which include multiplication we would prefer to show that the corresponding theory $\mathbf{VTC}^0$ can define multiplication and prove its properties, such as commutativity, rather than to assume commutativity as an axiom as typical single-sorted theories do.

It is not known whether any of the inclusions in (1) is proper, except for first two. Since all our basic theories have the same underlying language and use the same standard, Σ_1^1-definability, to compare their ability to define complex functions, it is easy to show that any two complexity classes in the top row of the table (2) are provably equal iff the corresponding theories have the same theorems.

However we are not just interested in which functions are Σ_1^1-definable in a theory, we are also interested in which universal combinatorial principles are provable. Since all true $\forall\Sigma_0^b$-sentences can be added to a single-sorted theory (or all true $\forall\Sigma_0^B$-sentences to a two-sorted theory) without increasing the class of Σ_1^1-definable functions, it is important to choose the axioms for the theory carefully. Our aim is to define "minimal canonical" theories for each complexity class. We justify this claim partly by defining, for most of the theories, a universally axiomatized conservative extension of the theory. The underlying language for each extension consists of function symbols for all the functions in the language, and the axioms consist of defining properties for the functions, such as recursion equations.

The prototype for these universal theories is the equational theory **PV** [14] and its single-sorted quantified counterpart **QPV** (also called PV_1), which has function symbols for the polynomial-time functions, and recursion equations based on Cobham's theorem for these functions as axioms. Here we present a two-sorted version **VPV** of **QPV**, and show that it is a conservative extension of our proposed minimal theory $\mathbf{TV}^0$ for **P**. The axioms of the universal theory **VPV** consist only of the recursion equations for function symbols coming from Cobham's theorem, in addition to axioms giving properties of strings and numbers which can be expressed in the weak class $\mathbf{AC^0}$. For example, we do not assume a derived property such as commutativity of multiplication. This supports our claim that $\mathbf{TV}^0$ and **VPV** are "minimal" theories for **P**.

This minimal philosophy really pays off when it comes time for propositional translations (Section 5). Since our language $\mathcal{L}_A^2$ has only trivial string functions as primitives, the proposition translations are very easy to describe, and the translations for the theories' axioms have trivial propositional proofs (no commutativity of multiplication, for example). This is in stark contrast to propositional translations for the standard single-sorted language (see [31], Section 9.2).

Most of the complexity classes and theories in (2) have associated propositional proof systems. For example, bounded-depth Frege systems are associated with $\mathbf{AC^0}$, Frege systems with $\mathbf{NC}^1$, and Extended Frege systems with **P**. An interesting part of propositional proof complexity is to determine the minimal system for which a given combinatorial principle has polynomial-size proofs. For example Buss [11] proved the surprising result that certain tautologies associated with the game of Hex have polynomial-size $\mathbf{TC^0}$-Frege proofs. But it seems clear that his argument shows a stronger result, namely that the principles expressed by the tautologies are theorems in the theory $\mathbf{VTC}^0$, from which his result would be a corollary. One of the aims of this paper to convince researchers to formulate such results as results on the power of our theories, rather than as results on the power of the associated propositional systems.

Some background in bounded arithmetic would be helpful in reading this paper. Good general references are [9, 10, 25, 31].

1.1 Organization

Section 2 presents the syntax and semantics of our two-sorted theories. The class $\mathbf{AC^0}$ and the notion of $\mathbf{AC^0}$ reducibility are defined. The theories $\mathbf{V^0}$ and $\overline{\mathbf{V^0}}$ are defined and studied. The definability of functions and relations in theories is studied.

Section 3 concerns single-sorted and two-sorted theories for polynomial time and the polynomial hierarchy. RSUV isomorphism between the two kinds of theories is explained. The theories **VPV** and $\mathbf{TV}^0$ for polynomial time are introduced and studied.

Section 4 presents theories for $\mathbf{TC^0}$, $\mathbf{AC^0}(2)$, $\mathbf{NC}^1$, **L**, **NL**, and **NC**.

Section 5 presents translations from our two-sorted theories to systems for the quantified propositional calculus, and discusses theories for **PSPACE**.

Section 6 gives some concluding remarks.

2 Two-sorted theories

2.1 Syntax and semantics

Our two-sorted theories are often called "second-order " theories, but their underlying logic is the two-sorted first-order predicate calculus, with equality for both sorts. They are based on the clean syntax of Zambella [48, 49]. The underlying language $\mathcal{L}_A^2$ has variables $x, y, z, \ldots$ for the first sort, called *number variables*, and variables $X, Y, Z, \ldots$ of the second sort, called *string variables*. The number variables are intended to range over $\mathbb{N}$, and the string variables are intended to range over finite subsets of $\mathbb{N}$ (which represent binary strings).

The language $\mathcal{L}_A^2$ includes the symbols $0, 1, +, \cdot$ of arithmetic on $\mathbb{N}$. In addition, we include the relation symbols $\leq, =$ for numbers, the function symbol $|X|$ on strings, the set membership relation $\in$, and string equality $=_2$. (In practice we will drop the subscript of $=_2$, since it will be clear from context.) The function $|X|$ denotes 1 plus the largest element in X, or 0 if X is empty (roughly the length of the corresponding string). We will use the notation $X(t)$ for $t \in X$, and we will think of $X(t)$ as the t-th bit in the string X.

Number terms are built from the constants 0,1, variables $x, y, z, \ldots$, and length terms $|X|$ using $+$ and $\cdot$. The only *string terms* are string variables $X, Y, Z, \ldots$. The atomic formulas are $\mathtt{T}$, $\mathtt{F}$, $t = u$, $X = Y$, $t \leq u$, $t \in X$ for any number terms t, u 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 $\exists X \leq t\ \phi$ stands for $\exists X(|X| \leq t \wedge \phi)$ and $\forall X \leq t\ \phi$ stands for $\forall X(|X| \leq t \supset \phi)$, where X does not occur in the term t.

A structure for $\mathcal{L}_A^2$ 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 of symbols of $\mathcal{L}_A^2$ is interpreted in $\langle U_1, U_2\rangle$ by a relation or function of appropriate type, with $=$ 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 $\mathcal{L}_A^2$ gets its intended interpretation.

$\Sigma_0^B = \Pi_0^B$ is the set of all formulas over $\mathcal{L}_A^2$ such that all number quantifiers are bounded, and there are no string quantifiers. (There may be free string variables.) For $i > 0$, Σ_i^B is defined recursively to be the set of all formulas beginning with a block of zero or more bounded existential string quantifiers followed by a Π_{i-1}^B formula, and Π_i^B is the set of all formulas beginning with a block of zero or more bounded universal string quantifiers followed by a Σ_{i-1}^B formula. Σ_1^1 is the set of formulas that begin with zero or more existential string quantifiers (bounded or unbounded), followed by a Σ_0^B formula.

If $\mathcal{L}$ is a language extending $\mathcal{L}_A^2$, then the formula classes $\Sigma_i^B(\mathcal{L})$ and $\Pi_i^B(\mathcal{L})$ are defined in the same way as Σ_i^B and Π_i^B, except we allow functions and predicates from $\mathcal{L}$,

$$\text{provided all terms bounding quantifiers are in the original language } \mathcal{L}_A^2. \tag{3}$$

Thus every bounded quantifier is bounded by a polynomial in $\vec{x}, |\vec{X}|$, where $\vec{x}, \vec{X}$ are the free variables in the formula.

We say that a formula is *bounded* if all of its quantifiers are bounded and satisfy the proviso (3).

Note that for $i > 1$ our Σ_i^B and Π_i^B formulas correspond to *strict* versions of the formula classes

$\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ (without #) defined in standard treatments because we require that all string quantifiers are in front. For example, the standard definition allows a formula to be in $\Sigma_1^{1,b}$ if it has a prenex form whose quantifier prefix is a mixture of bounded existential string quantifiers and bounded universal number quantifiers, and a similar definition applies to the single-sorted class Σ_1^b. In stronger two-sorted theories such as $V_1^i, i \geq 1$ the replacement scheme (20) (asserting that a formula beginning with $\forall x \leq a \exists Y \leq b$ is equivalent to one beginning $\exists Z \leq b \forall x \leq a$) holds, so every $\Sigma_1^{1,b}$ formula in the general sense is provably equivalent to one in the strict sense. However in the weaker theories considered here replacement does not hold in general (without surprising complexity-theoretic consequences [21]), so we stick with the simpler strict definition. This definition substantially simplifies the statements and proofs of witnessing theorems, since it eliminates the need for defining the formula $Witness_\phi^{i,\vec{x}}$ [4].

2.2 Two-sorted complexity classes

For a typical single-sorted theory of bounded arithmetic a member of the associated complexity class is a relation or function on $\mathbb{N}$. For our two-sorted complexity classes the relations $R(\vec{x}, \vec{Y})$ have both number arguments x_i ranging over $\mathbb{N}$ and string arguments Y_i ranging over finite subsets of $\mathbb{N}$. When the complexity class is defined in terms of machines or circuits, we assume that each number input is presented in unary notation (n is represented by a string of n 1's), and each finite subset (string) input is presented by the corresponding bit string. For example **P** is the class of such relations accepted in polynomial time on a Turing machine.

Our basic complexity class is $\mathbf{AC^0}$. Informally, a problem is in nonuniform $\mathbf{AC^0}$ if it can be accepted by a polynomial-size constant-depth family of circuits with unbounded fan-in OR and AND gates. However we shall refer to the uniform version, which has several equivalent definitions, including **LH** (the log time hierarchy on an alternating Turing machine) or **FO** (describable by a single-sorted formula using $<$ and Bit predicates [28]). In our two-sorted setting, we use alternating Turing machines to define $\mathbf{AC^0}$ as a class of relations, where number inputs are presented in unary and string inputs are presented in binary.

Definition 2.1 *A relation $R(\vec{x}, \vec{X})$ is in $\mathbf{AC^0}$ iff some alternating Turing machine accepts R in time $O(\log n)$ with a constant number of alternations.*

The following result ([28] and [17] pp 54–55) nicely connects $\mathbf{AC^0}$ and our two-sorted language $\mathcal{L}_A^2$.

Theorem 2.2 (Σ_0^B Representation Theorem) *A relation $R(\vec{x}, \vec{X})$ is in $\mathbf{AC^0}$ iff it is represented by some Σ_0^B formula $\phi(\vec{x}, \vec{X})$.*

Examples of $\mathbf{AC^0}$ relations are

$$Bit(X, i) \equiv X(i)$$
$$Reverse(X, Y) \equiv |X| = |Y| \wedge \forall x < |X| \forall y < |X| (x + y + 1 = |X| \supset (X(x) \leftrightarrow Y(y)))$$

In the second example the condition $|X| = |Y|$ guarantees, assuming $n = |X| > 0$, that $X(n-1) = Y(n-1) = 1$, and the bit sequence $X(0), ..., X(n-1)$ is the same as $Y(n-1), ..., Y(0)$.

A more interesting example is binary addition: $Plus(X, Y, Z) \equiv Z = X + Y$, where a string X represents the number $\sum_i 2^{X(i)}$. Then

$$Plus(X,Y,Z) \leftrightarrow |Z| \leq |X| + |Y| \wedge \forall i < |X| + |Y| [Z(i) \leftrightarrow X(i) \oplus Y(i) \oplus Carry(i,X,Y)] \quad (4)$$

where $\oplus$ is exclusive or, and

$$Carry(i,X,Y) \equiv \exists j < i[X(j) \wedge Y(j) \wedge \forall k < i(j < k \supset (X(k) \vee Y(k))] \quad (5)$$

On the other hand the relation $Parity(X)$, which holds iff X has an odd number of 1's, is *not* in $\mathbf{AC^0}$ [1, 24]. It follows that binary multiplication $Times(X, Y, Z) \equiv X \cdot Y = Z$ is not in $\mathbf{AC^0}$.

If $R(\vec{x})$ has no string arguments, then $\mathbf{AC^0}$ coincides with with the single-sorted complexity class **LTH** (Linear Time Hierarchy), which consists of relations on $\mathbb{N}$ recognizable by an alternating Turing machine in time linear in the *binary length* of its inputs, with a constant number of alternations. Thus for natural number arguments, the two-sorted log time hierarchy **LH** coincides with the single-sorted linear-time hierarchy, since in the first case numbers are presented in unary notation and in the second case numbers are presented in binary. However the main interest in two-sorted complexity classes is in string inputs, and the number inputs should be considered auxiliary.

Examples of numerical relations in (two-sorted) $\mathbf{AC^0}$ are $TimesNum(x, y, z) \equiv x \cdot y = z$ and $PrimeNum(x) \equiv x$ is a prime number.

For $i \geq 1$, the i-th level Σ_i^p of the (two-sorted) polynomial hierarchy can be defined as the class of relations $R(\vec{x}, \vec{X})$ accepted by an alternating Turing machine in time polynomial in the length of its input, with at most $i - 1$ alternations, beginning with existential. Thus $\mathbf{NP} = \Sigma_1^p$.

Theorem 2.3 (Σ_i^B Representation Theorem) *For $i \geq 1$, a relation $R(\vec{x}, \vec{X})$ is in Σ_i^p iff it is represented by some Σ_i^B formula $\phi(\vec{x}, \vec{X})$.*

See [9], page 106, for a proof in the single-sorted setting.

The two-sorted class **P** (polynomial time) and **NL** (nondeterministic log space) are also represented by syntactically-defined classes of formulas over $\mathcal{L}_A^2$, as we will see later in the paper.

2.3 Function classes and $\mathbf{AC^0}$-reducibility

Associated with each two-sorted complexity class **C** of relations is a two-sorted function class **FC**. Two-sorted functions are either *number functions* or *string functions*. A number function $f(\vec{x}, \vec{Y})$ takes values in $\mathbb{N}$, and a string function $F(\vec{x}, \vec{Y})$ takes finite subsets of $\mathbb{N}$ as values. A function f or F is *polynomially bounded* (or *p-bounded*) if there is a polynomial $p(\vec{x}, \vec{y})$ such that $f(\vec{x}, \vec{Y}) \leq p(\vec{x}, |\vec{Y}|)$ or $|F(\vec{x}, \vec{Y})| \leq p(\vec{x}, |\vec{Y}|)$. All function complexity classes we consider here contain only p-bounded functions.

Definition 2.4 (bit graph and FC) *The* bit graph B_F *of a string function F is defined by*

$$B_F(i, \vec{x}, \vec{Y}) \leftrightarrow F(\vec{x}, \vec{Y})(i)$$

If **C** *is a two-sorted complexity class of relations, then the corresponding functions class* **FC** *consists of all p-bounded number functions whose graphs are in* **C***, together with all p-bounded string functions whose bit graphs are in* **C**.

In particular, the string functions in $\mathbf{FAC^0}$ are those p-bounded functions whose bit graphs are in $\mathbf{AC^0}$. (Zambella [48] used $\mathcal{R}$ for $\mathbf{FAC^0}$ and called it the class of rudimentary functions. However there is danger here of confusion with Smullyan's rudimentary relations [44].) The nonuniform version of $\mathbf{FAC^0}$ consists of functions computable by bounded-depth polynomial-size circuits, and it is clear from this definition that the class is closed under composition. It follows from Corollary 2.10 below that the uniform class is also closed under composition.

Let $\mathcal{L}$ be a collection of two-sorted functions and relations. By this we mean a collection of relation symbols $R(\vec{x},\vec{X})$ of various arities together with string and number function symbols $f(\vec{x},\vec{X})$, $F(\vec{x},\vec{X})$ of various arities. We assume that each relation and function symbol in $\mathcal{L}$ has an intended interpretation in the standard structure $\underline{\mathbb{N}}_2$.

In the nonuniform setting, an $\mathbf{AC^0}$ reduction to $\mathcal{L}$ is defined by a polynomial size family of bounded-depth circuits, which is allowed NOT gates, unbounded fanin AND and OR gates, and gates which compute relations or functions in $\mathcal{L}$. In the uniform setting $\mathbf{AC^0}$ reducibility is defined by a log time constant alternation alternating Turing machine which is allowed oracle access to functions and relations in $\mathcal{L}$. Rather than giving a careful definition of the latter, we will give an equivalent definition in terms of Σ_0^B formulas.

Definition 2.5 *A string function $F(\vec{x},\vec{X})$ is* Σ_0^B-definable *from a collection $\mathcal{L}$ of two-sorted functions and relations if there is $\Sigma_0^B(\mathcal{L})$ formula $\phi(z,\vec{x},\vec{X})$ and a term $t = t(\vec{x},\vec{X})$ over $\mathcal{L}_A^2$ such that for all $z,\vec{x},\vec{X}$*

$$F(\vec{x},\vec{X})(z) \leftrightarrow z < t \wedge \phi(z,\vec{x},\vec{X}) \tag{6}$$

Similarly a number function $f(\vec{x},\vec{X})$ is Σ_0^B-definable *from $\mathcal{L}$ if there are such ϕ and t satisfying*

$$f(\vec{x},\vec{X}) = z \leftrightarrow z < t \wedge \phi(z,\vec{x},\vec{X})$$

We note that since the formulas ϕ in the above definition are allowed to have any terms built from functions in $\mathcal{L}$, it follows that any composition of functions in $\mathcal{L}$ is Σ_0^B-definable from $\mathcal{L}$.

Definition 2.6 *We say that $F(\vec{x},\vec{X})$ is $\mathbf{AC^0}$ reducible to $\mathcal{L}$ if there is a sequence $F_1,...,F_n$ of string functions such that $F_n = F$, and F_i is Σ_0^B-definable from $\mathcal{L}\cup\{F_1,...,F_{i-1}\}$ for $i = 1,...,n$. We say that a number function $f(\vec{x},\vec{X})$ is $\mathbf{AC^0}$ reducible to $\mathcal{L}$ if there are string functions $F_1,...,F_n$ which are $\mathbf{AC^0}$-reducible to $\mathcal{L}$ and f is Σ_0^B-definable from $\mathcal{L}\cup\{F_1,...,F_n\}$.*

The following result is straightforward.

Lemma 2.7 *A number function f is $\mathbf{AC^0}$ reducible to $\mathcal{L}$ iff*

$$f(\vec{x},\vec{X}) = |F(\vec{x},\vec{X})|$$

for some string function F which is $\mathbf{AC^0}$-reducible to $\mathcal{L}$.

We say that the collection $\mathcal{L}$ is *closed under Σ_0^B definitions* if every function which is Σ_0^B-definable from $\mathcal{L}$ is represented by a symbol in $\mathcal{L}$, and $\mathcal{L}$ is closed under $\mathbf{AC^0}$ reductions if every function $\mathbf{AC^0}$-reducible to $\mathcal{L}$ is represented by a symbol in $\mathcal{L}$.

The next result is immediate from the definitions involved.

Lemma 2.8 *$\mathcal{L}$ is closed under Σ_0^B definitions iff $\mathcal{L}$ is closed under* $\mathbf{AC^0}$ *reductions.*

Theorem 2.9 *Let $\mathcal{L}_A^2(\mathbf{FAC^0})$ denote any vocabulary which extends $\mathcal{L}_A^2$ and whose function symbols denote precisely $\mathbf{FAC^0}$ and whose relation symbols denote a subset of $\mathbf{AC^0}$. Then $\mathcal{L}_A^2(\mathbf{FAC^0})$ is closed under $\mathbf{AC^0}$ reductions.*

Proof. By the previous two lemmas it suffices to show that if a string function F is Σ_0^B-definable from functions in $\mathbf{FAC^0}$ then F is in $\mathbf{FAC^0}$. Suppose that F satisfies (6), where ϕ is $\Sigma_0^B(\mathbf{FAC^0})$. It suffices to show that every such ϕ is equivalent to a Σ_0^B formula ϕ'. First, we may assume that ϕ contains no relations other than those in $\mathcal{L}_A^2$ by replacing any occurrence of such a relation by its Σ_0^B defining formula. Also we may assume that ϕ contains no number function by appealing to Lemma 2.7. We now proceed by induction on the maximum nesting depth of string functions G in ϕ, where G is not in $\mathcal{L}_A^2$ (i.e. we do not count $|\ |$ in the nesting depth).

Consider a maximum depth occurrence of such a G in ϕ. It either has the form $|G(\vec{t},\vec{T})|$ or it is an atomic formula $G(\vec{t},\vec{T})(r)$, for some terms $\vec{t},\vec{T},r$. Since G is in $\mathbf{FAC^0}$, it follows from Definition 2.4 that G satisfies

$$G(\vec{y},\vec{Y})(z) \leftrightarrow z < u \wedge \psi(z,\vec{y},\vec{Y})$$

for some $\mathcal{L}_A^2$ term u and Σ_0^B formula ψ. Thus we may replace an occurrence $G(\vec{t},\vec{T})(r)$ by $r < u' \wedge \psi(r,\vec{t},\vec{T})$. On the other hand, an occurrence $|G(\vec{t},\vec{T})|$ in ϕ must be in the context $\alpha(|G(\vec{t},\vec{T})|)$ for some atomic formula α. The condition $w = |G(\vec{t},\vec{T})|$ can be expressed by $\eta(w,\vec{t},\vec{T})$ for some Σ_0^B-formula η, so $\alpha(|G(\vec{t},\vec{T})|)$ can be replaced by

$$\exists w \le u'(\eta(w,\vec{t},\vec{T}) \wedge \alpha(w))$$

Thus in either case we have reduced the number of occurrences of a function of maximal nesting depth in ϕ. We can proceed in this manner to get rid of them all. $\square$

Corollary 2.10 $\mathbf{FAC^0}$ *is closed under composition.*

2.4 The theory $\mathbf{V^0}$

Our base theory $\mathbf{V^0}$ [18, 17], called Σ_0^p-*comp* in [48] and $I\Sigma_0^{1,b}$ (without #) in [31], is associated with the complexity class $\mathbf{AC^0}$, and all two-sorted theories considered in this paper are extensions of $\mathbf{V^0}$. The language of $\mathbf{V^0}$ is $\mathcal{L}_A^2$. The axioms of $\mathbf{V^0}$ consist of the universal closures of the Σ_0^B formulas 2-*BASIC* together with the Σ_0^B comprehension scheme below. 2-*BASIC* consists of

B1. $x+1 \ne 0$	**B8.** $(x \le y \wedge y \le x) \supset x = y$
B2. $x+1 = y+1 \supset x = y$	**B9.** $0+1 = 1$
B3. $x+0 = x$	**B10.** $0 \le x$
B4. $x+(y+1) = (x+y)+1$	**B11.** $x \le y \wedge y \le z \supset x \le z$
B5. $x \cdot 0 = 0$	**B12.** $x \le y \vee y \le x$
B6. $x \cdot (y+1) = (x \cdot y)+x$	**B13.** $x \le y \leftrightarrow x < y+1$
B7. $x \le x+y$	**B14.** $x \ne 0 \supset \exists y \le x(y+1 = x)$
L1. $X(y) \supset y < \lvert X\rvert$	**L2.** $y+1 = \lvert X\rvert \supset X(y)$

SE. $X = Y \leftrightarrow [|X| = |Y| \wedge \forall i < |X|(X(i) \leftrightarrow Y(i))]$

The set 2-$BASIC$ of axioms is similar to the axioms for Zambella's theory Θ [48] and form the two-sorted analog of Buss's single-sorted axioms $BASIC$ [4]. Axioms **B1** to **B14** give basic properties of $0, 1, +, \cdot, \leq$, and **L1** and **L2** characterize $|X|$ to be one more than the largest element of X, or 0 if X is empty.

The Σ_0^B comprehension scheme is

$$\Sigma_0^B\text{-COMP:} \qquad \exists Z \leq y \forall i < y(Z(i) \leftrightarrow \phi(i, \vec{x}, \vec{X})) \tag{7}$$

where $\phi(i, \vec{x}, \vec{X})$ is any Σ_0^B formula not containing Z.

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

$$[X(0) \wedge \forall y < z(X(y) \supset X(y+1))] \supset X(z) \tag{8}$$

(See [18] or [17] for details.) From this and Σ_0^B-COMP we have

Theorem 2.11 $\mathbf{V^0}$ *proves the schemes*

$$\Sigma_0^B\text{-}IND\text{:} \qquad [\phi(0) \wedge \forall x(\phi(x) \supset \phi(x+1))] \supset \forall z \phi(z)$$

and

$$\Sigma_0^B\text{-}MIN\text{:} \qquad \exists x \phi(x) \supset \exists x[\phi(x) \wedge \neg \exists y(y < x \wedge \phi(y))]$$

where $\phi(x)$ is any Σ_0^B-formula (possibly containing parameters).

$\mathbf{V^0}$ proves simple properties of $\mathbf{AC^0}$ functions and predicates, such as the commutativity of string addition:

$$\mathbf{V^0} \vdash Plus(X, Y, Z) \leftrightarrow Plus(Y, X, Z)$$

This is because of the symmetric definition of $Plus$ (4), (5).

However $\mathbf{V^0}$ does not prove the pigeonhole principle, when the map from pigeons to holes is represented by a string variable X. This is shown using propositional proof lower bounds in a later section.

It is not hard to show that $\mathbf{V^0}$ is a conservative extension of the single-sorted theory $\mathbf{I\Delta_0}$ (Peano Arithmetic with induction restricted to bounded formulas [38]). That $\mathbf{V^0}$ is an extension of $\mathbf{I\Delta_0}$ is immediate from the 2-$BASIC$ axioms and Theorem 2.11. Conservativity follows from the fact that every model of $\mathbf{I\Delta_0}$ can be expanded to a model of $\mathbf{V^0}$ (the string universe consists of all Δ_0-definable sets from the number universe) [17].

The theory $\mathbf{I\Delta_0}$ (and hence $\mathbf{V^0}$) proves the basic properties of $+$ and $\cdot$ on $\mathbb{N}$, and allows us to define other number theoretic functions, including the binary length $|x|$ of a number x and the predicate $Bit(i, x)$, which holds iff the i-th bit (i.e. coefficient of 2^i of the binary notation of x is 1) and prove their basic properties [9, 25, 17].

The following result is proved in [18].

Theorem 2.12 *The theory* $\mathbf{V^0}$ *is finitely axiomatizable.*

2.5 Multidimensional arrays and the pigeonhole principle

We use $\langle x, y\rangle$ to abbreviate the term $(x+y)(x+y+1)+2y$. This is a pairing function, and $\mathbf{V^0}$ proves that the map $(x,y) \mapsto \langle x, y\rangle$ is a 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 $X(x,y) = X(\langle x,y\rangle)$. By iterating the pairing function we can define a multidimensional array $X(\vec{x})$. Then $\mathbf{V^0}$ proves the corresponding comprehension scheme.

Lemma 2.13 $\mathbf{V^0}$ *proves the k-ary Σ_0^B comprehension scheme*

$$\exists Z \leq \langle \vec{y}\rangle \forall \vec{x} < \vec{y}(Z(\vec{y}) \leftrightarrow \phi(\vec{x}, \vec{z}, \vec{X}))$$

for any Σ_0^B formula ϕ.

If we think of Z as a two-dimensional array, then we can represent row x in this array by $Z^{[x]}$ [48], where $G(x,Z) = Z^{[x]}$ is the $\mathbf{FAC^0}$ string function with bit-defining axiom

$$Z^{[x]}(i) \leftrightarrow i < |Z| \wedge Z(x,i) \tag{9}$$

(In the next section we point out that adding Σ_0^B-bit-defined functions to $\mathbf{V^0}$ provides a conservative extension over $\mathbf{V^0}$.)

As an example, we can formulate the pigeonhole principle PHP_n^{n+1} by a Σ_0^B formula $PHP(n,X)$ which asserts that if X codes a map from $n+1$ pigeons to n holes, then some hole has at least two pigeons. Thus $PHP(n,X)$ is

$$\forall i \leq n \exists j < n X(i,j) \supset \exists i \leq n \exists j \leq n \exists k < n(i < j \wedge X(i,k) \wedge X(j,k)) \tag{10}$$

It follows from the fact that the propositional version of PHP_n^{n+1} does not have polynomial size bounded-depth Frege proofs that $PHP(n,X)$ is not a theorem of $\mathbf{V^0}$ (see Section 5.3).

2.6 $\overline{\mathbf{V^0}}$: A universal conservative extension of $\mathbf{V^0}$

A universal formula is a formula in prenex form in which all quantifiers are universal. A universal theory is a theory which can be axiomatized by universal formulas. Here we introduce a universal theory $\overline{\mathbf{V^0}}$ which is a conservative extension of $\mathbf{V^0}$. The language $\mathcal{L}_{\mathbf{FAC^0}}$ of $\overline{\mathbf{V^0}}$ extends $\mathcal{L}_A^2$ by adding function symbols for all functions in $\mathbf{FAC^0}$. Each new function symbol has open defining axioms.

We need to introduce the predecessor function $pd(x)$ with defining axioms

B14′ $pd(0) = 0$ **B14″** $x \neq 0 \supset pd(x) + 1 = x$

in order to replace quantified axiom **B14**.

Now the idea is that for each open formula $\alpha(z, \vec{x}, \vec{X})$ and bounding term t there is a string function $F_{\alpha,t}$ in $\mathcal{L}_{\mathbf{FAC^0}}$ with bit graph α and a number function $f_{\alpha,t}$ satisfying

$$f_{\alpha,t}(\vec{x}, \vec{X}) = \min z < t\, \alpha(z, \vec{x}, \vec{X}) \tag{11}$$

where the RHS means the least number z such that $z < t$ and $\alpha(z, \vec{x}, \vec{X})$, or t if there is no such z.

More formally we give an inductive definition of the functions in $\mathcal{L}_{\mathbf{FAC^0}}$ and their defining axioms.

(a) $\mathcal{L}_{\mathbf{FAC^0}}$ includes $\mathcal{L}_A^2 \cup \{pd\}$.

(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

$$F_{\alpha,t}(\vec{x}, \vec{X})(z) \leftrightarrow z < t \wedge \alpha(z, \vec{x}, \vec{X}) \tag{12}$$

(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

$$f_{\alpha,t}(\vec{x}, \vec{X}) \leq t(\vec{x}, \vec{X}) \tag{13}$$
$$z < t \wedge \alpha(z, \vec{x}, \vec{X}) \supset \alpha(f_{\alpha,t}(\vec{x}, \vec{X}), \vec{x}, \vec{X})) \tag{14}$$
$$z < f_{\alpha,t}(\vec{x}, \vec{X}) \supset \neg\alpha(z, \vec{x}, \vec{X}) \tag{15}$$

We need to modify the axiom **SE** (extensionality) for $\mathbf{V^0}$ because it has a quantifier. The left-to-right direction of **SE** can be expressed by an open formula simply by replacing $\forall i < |X|$ by $i < |X| \supset$. The right-to-left direction of **SE** has an implicit quantifier $\exists i < |X|$. We can get rid of this by using the appropriate function $f_{\alpha,t}$ from (11) and then using Lemma 2.15 below. Let **SE**$'$ be the resulting quantifier-free version of **SE**.

Definition 2.14 $\overline{\mathbf{V^0}}$ *is the theory whose language is* $\mathcal{L}_{\mathbf{FAC^0}}$ *and whose axioms are the universal closures of* **B1**,...,**B13**, **B14**$'$,**B14**$''$,**L1**,**L2**,**SE**$'$, *and the defining axiom (12) for each function* $F_{\alpha,t}$ *and defining axioms (13,14,15) for each function* $f_{\alpha,t}$.

The following result is an easy consequence of the defining axioms for $f_{\alpha,t}$.

Lemma 2.15

$$\overline{\mathbf{V^0}} \vdash \exists z < t\alpha(z, \vec{x}, \vec{X}) \;\leftrightarrow\; f_{\alpha,t}(\vec{x}, \vec{X}) < t \wedge \alpha(f_{\alpha,t}(\vec{x}, \vec{X}), \vec{x}, \vec{X}))$$

Lemma 2.16 *For every* Σ_0^B *formula* ϕ *there is an open formula* α *of* $\mathcal{L}_{\mathbf{FAC^0}}$ *such that* $\overline{\mathbf{V^0}}$ *proves* $(\phi \leftrightarrow \alpha)$. *For every open formula* α *of* $\mathcal{L}_{\mathbf{FAC^0}}$ *there is a* Σ_0^B *formula* ϕ *such that* $\overline{\mathbf{V^0}}$ *proves* $(\phi \leftrightarrow \alpha)$.

Proof. The first sentence follows by structural induction on Σ_0^B formulas ϕ, using Lemma 2.15. 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 Σ_0^B formula ϕ. Use an argument similar to the proof of Theorem 2.9. □

Theorem 2.17 $\overline{\mathbf{V^0}}$ *is a universal theory which 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. From the first sentence of Lemma 2.16 and (12) we have that for every Σ_0^B formula $\phi(i, \vec{x}, \vec{X})$ there is a function F in $\mathcal{L}_{\mathbf{FAC^0}}$ such that

$$\overline{\mathbf{V^0}} \vdash \quad F(\vec{x}, \vec{X})(i) \leftrightarrow i < y \wedge \phi(i, \vec{x}, \vec{X})$$

It follows that $\overline{\mathbf{V^0}}$ proves the Σ_0^B-COMP axioms (7). Since **B14″** implies **B14**, we conclude that $\overline{\mathbf{V^0}}$ is an extension of $\mathbf{V^0}$.

To see that the extension is conservative, we use the second sentence of Lemma 2.16 to show that every model of $\mathbf{V^0}$ has a unique expansion to a model of $\overline{\mathbf{V^0}}$: we use (12) and Σ_0^B-COMP in the case of string functions $F_{\alpha,t}$ and the Σ_0^B-MIN scheme and (11) in the case of a number functions $f_{\alpha,t}$.

Finally it is clear from the definition 2.4 of $\mathbf{FAC^0}$ that the function symbols in $\mathcal{L}_{\mathbf{FAC^0}}$ represent precisely the functions in $\mathbf{FAC^0}$. $\square$

It is worth emphasizing that $\overline{\mathbf{V^0}}$ proves the Σ_0^B-IND and Σ_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.

2.7 Witnessing

The idea of replacing an existential quantifier in a formula by a function, so that for example $\forall x \exists y \phi(x,y)$ is replaced by $\forall x \phi(x, f(x))$, goes back to Skolem. Buss [4] studied the complexity of finding such a "witness" $f(x)$ for $\exists y$ in case $\forall x \exists y \phi(x,y)$ is provable in certain theories. Since our notion of Σ_1^1 and Σ_1^B would be called "strict" in the usual terminology, we can simplify the standard definition of witnessing by avoiding mention of the somewhat complicated formulas $Witness_\phi^{i,\vec{x}}$.

Definition 2.18 *Let T be a theory over a language $\mathcal{L}$ which includes $\mathcal{L}_A^2$ and let $\phi(\vec{x}, \vec{X}, \vec{Y})$ be a $\Sigma_0^B(\mathcal{L})$-formula. Then functions $\vec{F}$ in $\mathcal{L}$* witness *the formula $\exists \vec{Y} \phi(\vec{x}, \vec{X}, \vec{Y})$ in T if $T \vdash \phi(\vec{x}, \vec{X}, \vec{F}(\vec{x}, \vec{X}))$*

Theorem 2.19 (**Witnessing**) *Suppose T is a universal theory which extends $\mathbf{V^0}$, and is defined over a language $\mathcal{L}$ and suppose that for every open formula $\alpha(i, \vec{x}, \vec{X})$ and term $t(\vec{x}, \vec{X})$ over $\mathcal{L}$ there is a function symbol F in $\mathcal{L}$ such that*

$$T \vdash F(\vec{x}, \vec{X})(i) \leftrightarrow i < t \wedge \alpha(i, \vec{x}, \vec{X})$$

Then every theorem of T of the form $\exists \vec{Y} \alpha(\vec{x}, \vec{X}, \vec{Y})$, where α is open, is witnessed in T by functions in $\mathcal{L}$.

Proof. To simplify notation assume a single existential quantifier, so T proves $\exists Y \alpha(\vec{x}, \vec{X}, Y)$. Since T is universal we may apply the Herbrand Theorem [10] to conclude there are string terms $R_1(\vec{x}, \vec{X}), ..., R_k(\vec{x}, \vec{X})$ such that

$$T \vdash (\alpha(\vec{x}, \vec{X}, R_1) \vee ... \vee \alpha(\vec{x}, \vec{X}, R_k))$$

The idea is to define the witnessing function $F(\vec{x}, \vec{X})$ to be the first term $R_i(\vec{x}, \vec{X})$ which satisfies α. Let

$$t(\vec{x}, \vec{X}) \equiv |R_1(\vec{x}, \vec{X})| + ... + |R_k(\vec{x}, \vec{X})|$$

Then F has defining formula (sometimes suppressing $\vec{x}, \vec{X}$)

$$F(\vec{x}, \vec{X})(z) \leftrightarrow z < t \wedge \bigvee_{i=1}^{k} (R_i(\vec{x}, \vec{X})(z) \wedge \alpha(R_i) \wedge \neg\alpha(R_1) \wedge ... \wedge \neg\alpha(R_{i-1}))$$

□

Corollary 2.20 *Every Σ_1^1 theorem of $\overline{\mathbf{V^0}}$ is witnessed in $\overline{\mathbf{V^0}}$ by functions in $\mathcal{L}_{\mathbf{FAC^0}}$.*

Proof. This follows from the Theorem 2.19 and Lemma 2.16. □

2.8 Bounded theories and definable functions

Definition 2.21 *A* p-bounded theory *T is an extension of $\mathbf{V^0}$ which has an axiomatization in which all axioms are universal closures of bounded formulas, where the quantifier bounds satisfy the proviso (3), and for each number function $f(\vec{x}, \vec{X})$ and each string function $F(\vec{x}, \vec{X})$ there is a term $t(\vec{x}, \vec{X})$ of $\mathcal{L}_A^2$ such that $T \vdash |F(\vec{x}, \vec{X})| \leq t(\vec{x}, \vec{X})$ and $T \vdash f(\vec{x}, \vec{X}) \leq t(\vec{x}, \vec{X})$.*

For example $\mathbf{V^0}$ and $\overline{\mathbf{V^0}}$ are p-bounded theories.

The following result is a two-sorted version of Parikh's theorem [38]. See [31, 25, 9] for proofs of single-sorted versions, or [17] for a two-sorted proof.

Theorem 2.22 *(Parikh) If T is a p-bounded theory and $T \vdash \exists \vec{Y} \phi$ where ϕ is a bounded formula, then there are terms $\vec{t}$ of $\mathcal{L}_A^2$ involving the free variables of ϕ, (but not involving $\vec{Y}$), such that $T \vdash \exists \vec{Y} < \vec{t} \phi$.*

Now we turn to the question of function definability in a theory. Here we select a class Φ of formulas to represent the graphs of functions, and consider a function Φ-definable in T if some ϕ in Φ represents the graph of the function and T proves that the relation represented by ϕ is total and single-valued. Thus

Definition 2.23 *Let T be an extension of $\mathbf{V^0}$ and let Φ be a class of formulas in the language of T. A string function $F(\vec{x}, \vec{X})$ is* Φ-definable *in T if it satisfies*

$$Y = F(\vec{x}, \vec{X}) \leftrightarrow \phi(\vec{x}, \vec{X}, Y) \tag{16}$$

for some formula ϕ in Φ, and

$$T \vdash \forall \vec{x} \forall \vec{X} \exists ! Y \phi(\vec{x}, \vec{X}, Y) \tag{17}$$

The Φ-definability for a number function $f(\vec{x}, \vec{X})$ is defined similarly.

Lemma 2.24 *If T is an extension of $\mathbf{V^0}$ which satisfies (17) and F is not in the language of T and T' is the result of adding F to the language and adding (16) as an axiom, then F' is a conservative extension of T.*

Proof. According to (17), every model of T has an expansion to a model of T' which satisfies (16). □

But how do we choose the class Φ of formulas in Definition 2.23? In the context of (unbounded) single-sorted arithmetic, the Σ_1 definable functions in a theory T might be referred to as the provably total recursive functions [9]. In the two-sorted setting this would correspond to the Σ_1^1-definable functions. In fact, if T is the theory of all true sentences over $\mathcal{L}_A^2$, then a total string or number function is Σ_1^1-definable in T iff it is computable (see for example [10]).

Thus we will use Σ_1^1-definability as our standard notion of definability in this paper. For each of the complexity classes C in (1) we will give a p-bounded theory T such that FC contains precisely the Σ_1^1-definable functions in T.

The following result is important for p-bounded theories, and follows immediately from Parikh's Theorem above.

Corollary 2.25 *If T is a p-bounded theory, then a function (string or number) is Σ_1^1-definable in T iff it is Σ_1^B-definable in T.*

The next result connects the theory $\mathbf{V^0}$ with the complexity class $\mathbf{FAC^0}$. An example of a string function that is Σ_1^1-definable in $\mathbf{V^0}$ is string addition $FPlus(X,Y)$. We use the formula $Plus(X,Y,Z)$ (4,5) to represent the graph of $FPlus$. Then $\mathbf{V^0}$ proves the existence of Z by Σ_0^B-COMP, and uniqueness is straightforward.

Theorem 2.26 *A function (string or number) is Σ_1^1-definable in $\mathbf{V^0}$ iff it is in $\mathbf{FAC^0}$.*

Proof. That every function in $\mathbf{FAC^0}$ is Σ_1^1-definable in $\mathbf{V^0}$ follows from Theorem 2.2 and Definition 2.4 of $\mathbf{FAC^0}$, using the axiom scheme Σ_0^B-COMP (7). The converse follows from Corollary 2.20, Theorem 2.17. □

The notions of Δ_1-definability and Δ_1^B-definability of relations correspond to Σ_1^1-definability and Σ_1^B-definability of functions, and are defined as follows.

Definition 2.27 *A relation $R(\vec{x},\vec{X})$ is Δ_1^1-definable in a theory T over a language $\mathcal{L}$ containing $\mathcal{L}_A^2$ if there are $\Sigma_1^1(\mathcal{L})$ formulas ϕ and ψ such that R satisfies*

$$R(\vec{x},\vec{X}) \leftrightarrow \phi(\vec{x},\vec{X})$$

and T proves $(\phi(\vec{x},\vec{X}) \leftrightarrow \neg\psi(\vec{x},\vec{X}))$. Δ_1^B-definability is the same, except replace $\Sigma_1^1(\mathcal{L})$ by $\Sigma_1^B(\mathcal{L})$.

It follows from Parikh's Theorem (2.22) that if T is a p-bounded theory, then a relation R is Δ_1^1-definable in T iff it is Δ_1^B-definable in T.

For the next result, we define the characteristic function f_R of a relation R in the usual way: $f_R(\vec{x},\vec{X}) = 1$ if $R(\vec{x},\vec{X})$, and $f_R(\vec{x},\vec{X}) = 0$ otherwise.

Theorem 2.28 *If the language of a theory T includes $\mathcal{L}_A^2$, and a complexity class $\mathbf{C}$ has the property that for all relations R, $R \in \mathbf{C}$ iff $f_R \in \mathbf{FC}$, and the class of Σ_1^1-definable functions in T coincides with $\mathbf{FC}$, then the class of Δ_1^1-definable relations in T coincides with $\mathbf{C}$.*

Proof. Assume the hypotheses of the theorem, and suppose that $R(\vec{x},\vec{X})$ is Δ_1^1-definable in T. Then there are Σ_0^B formulas α and β such that

$$R(\vec{x},\vec{X}) \leftrightarrow \exists\vec{Y}\,\alpha(\vec{x},\vec{X},\vec{Y})$$

and

$$T \vdash (\exists \vec{Y}\, \alpha(\vec{x}, \vec{X}, \vec{Y}) \leftrightarrow \neg \exists \vec{Y}\, \beta(\vec{x}, \vec{X}, \vec{Y})) \tag{18}$$

Thus the characteristic function $f_R(\vec{x}, \vec{X})$ of R satisfies

$$y = f_R(\vec{x}, \vec{X}) \leftrightarrow \phi(y, \vec{x}, \vec{X}) \tag{19}$$

where

$$\phi(y, \vec{x}, \vec{X}) \equiv \exists \vec{Y}(y = 1 \wedge \alpha(\vec{x}, \vec{X}, \vec{Y}) \vee y = 0 \wedge \beta(\vec{x}, \vec{X}, \vec{Y}))$$

Then T proves $\exists ! y \phi(y, \vec{x}, \vec{X})$, where existence of y and $\vec{Y}$ follows from the $\leftarrow$ direction of (18) and uniqueness of y follows from the $\rightarrow$ direction of (18). Thus f_R is Σ_1^1-definable in T, so f_R is in **FC**, so R is in **C**.

Conversely, suppose that $R(\vec{x}, \vec{X})$ is in **C**, so f_R is in **FC**. Then f_R is Σ_1^1-definable in T, so there is a Σ_1^1 formula $\phi(y, \vec{x}, \vec{X})$ such that (19) holds and

$$T \vdash \exists ! y \phi(y, \vec{x}, \vec{X})$$

Then $R(\vec{x}, \vec{X}) \leftrightarrow \exists y(y \neq 0 \wedge \phi(y, \vec{x}, \vec{X}))$ and

$$T \vdash \quad \exists y(y \neq 0 \wedge \phi(y, \vec{x}, \vec{X})) \leftrightarrow \neg \phi(0, \vec{x}, \vec{X})$$

so R is Δ_1^1-definable in T. $\square$

2.9 The replacement scheme

It is easy to show the following partial converse to Theorem 2.28: If T is a p-bounded theory and the Δ_1^1-definable relations in T coincide with **C**, then every function that is Σ_1^B-definable in T is in **FC**. But we need stronger assumptions to conclude that every function in **FC** is Σ_1^B-definable in T.

A sufficient condition is that T proves the Σ_1^B-replacement scheme below. However none of the minimal theories for any of the complexity classes up to **P** proves this scheme, unless there are unexpected consequences [21] (although $\mathbf{V}^1$ proves the scheme). This fact means that we must work harder to prove that each of these weaker theories Σ_1^B-defines the functions in its associated complexity class.

The Φ- *replacement scheme* is

$$\forall x < b \exists Z < b\, \phi(x, Z) \supset \exists W \forall x < b\, \phi(x, W^{[x]}) \tag{20}$$

where ϕ is in Φ (and may have parameters).

(See (9) for the notation $W^{[x]}$.)

3 Theories for P and PH

3.1 Single-sorted theories

Cook [14] (see also [22, 15]) introduced a single-sorted equational theory **PV** which has function symbols for every polynomial time function on $\mathbb{N}$. Each function symbol is introduced in a manner based on Cobham's [13] characterization of the polynomial time functions as the least class including certain initial functions and closed under composition and bounded recursion on notation (see Theorem 3.5 below). The initial functions include

$$x\#y = 2^{|x|\cdot|y|}$$

in order to give the functions polynomial growth rate (i.e. the length of the value is polynomial in the length of the arguments). The axioms consist of defining equations for each function symbol, and **PV** has a rule of inference that implements induction on binary notation.

The theory **QPV** [15] ('Q' for "quantified"; called PV_1 in [31]) is a single-sorted universal first-order theory over the language of **PV**, axiomatized by the theorems of **PV**. The theory **QPV** proves induction on notation for open formulas, but what is more surprising, **QPV** proves full induction for open formulas.

Theorem 3.1 **QPV** *proves the IND scheme*

$$[\phi(0) \wedge \forall x(\phi(x) \supset \phi(x+1))] \supset \forall z\phi(z) \tag{21}$$

for open formulas $\phi(x)$ (with parameters).

Proof. (sketch) Suppose that $\phi(x) = \phi(x, \vec{y})$ has parameters $\vec{y}$. Using the "Dowd/Statman trick", **QPV** can use binary search to define a polynomial time function g such that **QPV** proves

$$(\phi(0) \wedge \neg\phi(x)) \supset ((\phi(g(x,\vec{y})) \wedge \neg\phi(g(x,\vec{y})+1))$$

from which the induction formula follows. □

Buss [4] introduced a hierarchy $\mathbf{S}^1_2 \subseteq \mathbf{T}^1_2 \subseteq \mathbf{S}^2_2 \subseteq \mathbf{T}^2_2 \subseteq ...$ of theories for the polynomial hierarchy as follows.

The underlying language has symbols $[0, S, +, \cdot, \#, |x|, \lfloor\frac{1}{2}x\rfloor; =, \leq]$. The axioms include a set BASIC of 32 axioms giving simple properties of these symbols. In addition, the theory $\mathbf{S}^i_2$ has the Σ^b_i-PIND axiom scheme

$$[\phi(0) \wedge \forall x(\phi(\lfloor\frac{1}{2}x\rfloor) \supset \phi(x))] \supset \forall x\phi(x)$$

and $\mathbf{T}^i_2$ has the IND axiom scheme (21) for Σ^b_i formulas ϕ.

A *sharply bounded* quantifier is one of the form $\exists x < |t|$ or $\forall x < |t|$. A formula all of whose quantifiers are sharply bounded is a Σ^b_0 formula. In general, a Σ^b_i formula is one which has a prenex form beginning with zero or more bounded existential quantifiers followed by at most $i-1$ blocks of bounded quantifiers alternating between universal and existential, except any number of sharply bounded quantifiers may be mixed in. A Π^b_i formula is defined analogously.

Buss proved that the Σ_i^b-definable functions in $\mathbf{S}_2^i$ are those reducible to Σ_{i-1}^p. In particular, the Σ_1^b-definable (and the Σ_1-definable) functions in $\mathbf{S}_2^1$ are the polynomial time functions. Buss also proved that $\mathbf{S}_2^1$ is Σ_1^b-conservative over **QPV**. However $\mathbf{S}_2^1$ is not Σ_2^b-conservative over **QPV** unless the polynomial hierarchy collapses [33].

3.2 Two-sorted theories and RSUV isomorphism

Buss [4] introduced the two-sorted theories $\mathbf{U}_j^i$ and $\mathbf{V}_j^i$ to capture the complexity classes polynomial space, exponential time, and beyond, with respect to arguments of type number. Each of these theories has several variations, which are more or less equivalent for proving bounded theorems. Razborov [42] concentrated on $\mathbf{V}_1^1$ and its subtheories and argued that these theories are appropriate for formalizing results in Boolean circuit complexity, when circuits and Boolean function graphs are represented by objects of type string.

Here we present the theory $\mathbf{V}^i$, which is our version of $\mathbf{V}_1^i$ as an extension of $\mathbf{V^0}$.

Definition 3.2 *The theory $\mathbf{V}^i$ has the same language and axioms as $\mathbf{V^0}$ (Section 2.4) except the Σ_0^B comprehension scheme is replaced by the Σ_i^B comprehension scheme.*

Thus $\mathbf{V}^i$ proves the Σ_i^B-IND and Σ_i^B-MIN axioms. Also $\mathbf{V}^i$ proves the Σ_i^B-replacement scheme (9).

We use **P** to denote the class of two-sorted relations computable in polynomial time on a Turing machine, according to our standard input convention (Section 2.2) that number inputs are presented in unary and string inputs are presented as bit strings. Then **FP** denotes the corresponding function class according to Definition 2.4: a string function is in **FP** iff it is p-bounded and its bit graph is in **P**. It is easy to check that a two-sorted function is in **FP** iff it is computable in polynomial time on a Turing machine.

Theorem 3.3 *A function (string or number) is Σ_1^1-definable in $\mathbf{V}^1$ iff it is in* **FP**.

Proof. (sketch) Suppose that M is a Turing machine which computes a function $F(\vec{x},\vec{X})$ in polynomial time. Using a suitable coding of Turing machine computations it is possible to find a Σ_0^B formula ϕ_M such that $\phi_M(\vec{x},\vec{X},Y,Z)$ holds iff Z codes a halting computation of M on input $\vec{x},\vec{X}$ for which M outputs Y. Thus for some term $t = t(\vec{x},\vec{X})$

$$Y = F(\vec{x},\vec{X}) \leftrightarrow \exists Z < t\, \phi_M(\vec{x},\vec{X},Y,Z)$$

Further, $\mathbf{V}^1$ proves $\forall\vec{x}\forall\vec{X}\exists!Y\exists Z\phi_M(\vec{x},\vec{X},Y,Z)$. The existence of the computation Z is proved by induction on the number of steps, using Σ_1^B-IND. Uniqueness is proved by showing that the computation is unique. Thus F is Σ_1^1-definable in $\mathbf{V}^1$.

Conversely it is possible to prove a witnessing theorem stating that every Σ_1^1 theorem of $\mathbf{V}^1$ can be witnessed by functions in **FP**. However this theorem is more difficult than Theorem 2.19 because the Σ_1^B-COMP axioms are Σ_2^B formulas rather than Σ_1^B formulas. Witnessing can be proved by a cut-elimination argument [4, 17] or a model-theoretic argument [25, 48]. □

It turns out that the single-sorted theory $\mathbf{S}_2^i$ is essentially equivalent to the two-sorted theory $\mathbf{V}^i$, and in fact there is an "RSUV isomorphism" [30, 41, 47, 31] between the two theories. Model theoretically, there is a bijection between (isomorphism types of) models of $\mathbf{S}_2^i$ and models of $\mathbf{V}^i$. Each model $\mathcal{M}_1$ of $\mathbf{S}_2^i$ determines a model $\mathcal{M}_2$ of $\mathbf{V}^i$ whose string universe is the universe M_1 of $\mathcal{M}_1$ and whose number universe is the subset $\log(M_1) = \{|u| : u \in M_1\}$ of M_1. Each model $\mathcal{M}_2$ of $\mathbf{V}^i$ determines a model $\mathcal{M}_1$ of $\mathbf{S}_2^i$ whose universe is the string universe M_2 of $\mathcal{M}_2$. The maps have the property that if we go from $\mathcal{M}_1$ to $\mathcal{M}_2$ and back to $\mathcal{M}'_1$, then $\mathcal{M}_1$ is isomorphic to $\mathcal{M}'_1$. Similarly, if we go from $\mathcal{M}_2$ to $\mathcal{M}_1$ and then back to $\mathcal{M}'_2$, then $\mathcal{M}_2$ is isomorphic to $\mathcal{M}'_2$.

Syntactically, for $i \geq 1$, a Σ_i^b formula ϕ of $\mathbf{S}_2^i$ is provably equivalent to a strict Σ_i^b formula (meaning a prenex form in which all sharply bounded quantifiers come after the prefix of i blocks of bounded quantifiers), which translates to Σ_i^B formula $\phi^\sharp$ of $\mathbf{V}^i$ by giving Σ_1^B definitions of the function symbols of $\mathbf{S}_2^i$ as functions on strings. This must be done in such a way that the translations of the axioms of $\mathbf{S}_2^i$ are theorems in $\mathbf{V}^i$. The most work here is giving a definition of string multiplication $X \cdot Y$ and showing that $\mathbf{V}^i$ proves commutativity of multiplication.

Conversely a Σ_i^B formula ψ translates to a Σ_i^b formula $\psi^\flat$ by translating $i \in X$ by a formula asserting that bit i of x is 1, where X translates to x. It is straightforward to show that the translations of the axioms of $\mathbf{V}^i$ are theorems of $\mathbf{S}_2^i$, except proving the translations of the Σ_i^B comprehension scheme requires an interesting idea (using the numones function) [7]. The translations are such that $\mathbf{S}_2^i \vdash \phi^{\sharp\flat} \leftrightarrow \phi$ and $\mathbf{V}^i \vdash \psi^{\flat\sharp} \leftrightarrow \psi$.

Takeuti [47] introduced the name *RSUV isomorphism* to indicate equivalences respectively between the single-sorted hierarchies R and S and the two-sorted hierarchies U and V. Thus R_{k+1}^i is isomorphic to U_k^i, and S_{k+1}^i is isomorphic to V_k^i (see [31]).

3.3 The theory VPV

We introduce a two-sorted version of **QPV** called **VPV**. This is a universal theory with function symbols for all functions in **FP** and with axioms giving recursion equations for the functions, based on Cobham's theorem. We start by presenting a two-sorted version of Cobham's theorem.

The function $Chop(x, X)$ returns the initial segment of X of length x. It has the bit-defining axiom

$$Chop(x, X)(i) \leftrightarrow i < x \wedge X(i)$$

We sometimes write $X^{<x}$ for $Chop(x, X)$.

Definition 3.4 *A string function F is defined from G and H by* p-bounded recursion *if it satisfies the defining equations*

$$F(0, \vec{y}, \vec{Z}) = G(\vec{y}, \vec{Z}) \quad (22)$$

$$F(x+1, \vec{y}, \vec{Z}) = (H(x, \vec{y}, \vec{Z}, F(x, \vec{y}, \vec{Z})))^{<t(x,\vec{y},\vec{Z})} \quad (23)$$

for some term $t(x, \vec{y}, \vec{Z})$ of $\mathcal{L}_A^2$.

Theorem 3.5 (Two-sorted Cobham's Theorem) *A function (number or string) is in* **FP** *iff it can be obtained from functions in* $\mathbf{FAC^0}$ *by finitely many applications of composition and p-bounded recursion.*

Proof. (sketch) For the $\Longleftarrow$ direction, it is clear that $\mathbf{FAC^0} \subseteq \mathbf{FP}$, and it is easy to see that **FP** is closed under the operations of composition and p-bounded recursion.

For the $\Longrightarrow$ direction, suppose that M is a polynomial time Turing machine computing the function $F(\vec{x}, \vec{Y})$. It is possible to code the configurations of M by strings in such a way that the function

$$G(\vec{x}, \vec{Y}) = \text{ the initial configuration of } M \text{ on input } (\vec{x}, \vec{Y})$$

is in $\mathbf{FAC^0}$, and also the step function $Step(Z)$ is in $\mathbf{FAC^0}$, where if Z is any code for a configuration of M, then $Step(Z) = Z'$ and Z' is the configuration resulting in one step from Z. Now define

$$H(\vec{x}, \vec{Y}, Z) = Step(Z)$$

Let $Comp(i, \vec{x}, \vec{Y})$ be defined from G and H by limited recursion with bounding term t, where $t(i, \vec{x}, \vec{Y})$ is an upper bound on the length of the configuration of M on input $(\vec{x}, \vec{Y})$ after i steps. Then $Comp(i, \vec{x}, \vec{Y})$ is the configuration of M on input $(\vec{x}, \vec{Y})$ after i steps. Let $Out(Z)$ be the $\mathbf{FAC^0}$ function picking out the output string of M where Z is a final configuration of M. Then

$$F(\vec{x}, \vec{Y}) = Out(Comp(u(\vec{x}, \vec{Y}), \vec{x}, \vec{Y}))$$

where $u(\vec{x}, \vec{Y})$ is an upper bound on the computation time of M on input $(\vec{x}, \vec{Y})$. $\square$

The language $\mathcal{L}_{\mathbf{FP}}$ extends $\mathcal{L}_{\mathbf{FAC^0}}$ and has function symbols for all functions in $\mathbf{FP}$. We give an inductive definition of the functions in $\mathcal{L}_{\mathbf{FP}}$, in the same style as given for $\mathcal{L}_{\mathbf{FAC^0}}$ in Section 2.6.

(a), (b′), (c′) are the same as (a), (b), and (c) in Section 2.6 except $\mathcal{L}_{\mathbf{FAC^0}}$ is replaced by $\mathcal{L}_{\mathbf{FP}}$.

(d) For each triple G, H, t, where $G(\vec{y}, \vec{Z})$ and $H(\vec{y}, \vec{Z}, W)$ are functions in $\mathcal{L}_{\mathbf{FP}}$ and $t = t(x, \vec{y}, \vec{Z})$ is a term in $\mathcal{L}_A^2$ there is a function $F = F_{G,H,t}$ in $\mathcal{L}_{\mathbf{FP}}$ with defining equations (22,23)

Definition 3.6 **VPV** *is the theory whose language is* $\mathcal{L}_{\mathbf{FP}}$ *and whose axioms are the universal closures of* **B1**,...,**B13**, **B14′**,**B14″**,**L1**,**L2**,**SE′**, *and the defining axiom (12) for each function* $F_{\alpha,t}$ *and defining axioms (13,14,15) for each function* $f_{\alpha,t}$, *and defining axioms (22,23) for each* $F_{G,H,t}$.

Thus **VPV** is a universal theory and, by Cobham's Theorem 3.5, the functions symbols of **VPV** represent precisely the polynomial time functions **FP**.

Similar to Lemma 2.16, for every $\Sigma_0^B(\mathcal{L}_{\mathbf{FP}})$ formula ϕ there is an open formula α of $\mathcal{L}_{\mathbf{FP}}$ such that $\mathbf{VPV} \vdash (\phi \leftrightarrow \alpha)$. The next result follows immediately from this and Theorem 2.19.

Theorem 3.7 (Witnessing in VPV) *Every* $\Sigma_1^1(\mathcal{L}_{\mathbf{FP}})$ *theorem of* **VPV** *is witnessed in* **VPV** *by functions in* $\mathcal{L}_{\mathbf{FP}}$.

From this and Cobham's Theorem 3.5 we have

Theorem 3.8 **FP** *is the class of* $\Sigma_1^1(\mathcal{L}_{\mathbf{FP}})$*-definable functions in* **VPV**.

One can show that **QPV**, as defined in [15], is RSUV isomorphic to **VPV**, using techniques in [41, 47, 26].

3.4 The theories $\mathbf{TV}^0$ and $\mathbf{V}_1$-Horn

The theory $\mathbf{V}_1$-**Horn** [18] is a finitely axiomatizable extension of $\mathbf{V^0}$, over the language $\mathcal{L}_A^2$. It has the same axioms as $\mathbf{V^0}$, except Σ_0^B-COMP is replaced by a comprehension scheme over the class Σ_1^B-Horn, a class of formulas which represent precisely the relations in $\mathbf{P}$. In [18] it is shown that the Σ_1^B-definable functions in $\mathbf{V}_1$-**Horn** are exactly those in the class $\mathbf{FP}$, and that $\mathbf{QPV}$ is RSUV isomorphic to $\mathbf{V}_1$-**Horn**.

Here we introduce the theory $\mathbf{TV}^0$, which turns out to have the same theorems as $\mathbf{V}_1$-**Horn**, but is defined in the style of Buss's hierarchy $\mathbf{T}_2^i$. We view a string X as the number $\sum_i 2^{X(i)}$ and introduce the string successor function $S(X)$ by the Σ_0^B bit definition

$$S(X)(i) \leftrightarrow \phi_S^{bit}(i, X) \tag{24}$$

where

$$\phi_S^{bit}(i,X) \;\equiv\; i \le |X| \wedge [(X(i) \wedge \exists j < i \neg X(j)) \vee (\neg X(i) \wedge \forall j < i X(j))]$$

We use $\mathbf{0}$ to denote the empty string (with bit defining axiom $\mathbf{0}(i) \leftrightarrow 0 \neq 0$).

The Σ_0^B-String-IND scheme is

$$[\phi(\mathbf{0}) \wedge \forall X(\phi(X) \supset \phi(S(X))] \;\supset\; \phi(Y) \tag{25}$$

where $\phi(X)$ is Σ_0^B.

Since we want the theory $\mathbf{TV}^0$ to have underlying language $\mathcal{L}_A^2$, we will interpret (25) as a formula over $\mathcal{L}_A^2$ using the standard construction showing that introduction of a Σ_0^B bit-definable function results in a conservative extension of $\mathbf{V^0}$.

Definition 3.9 *The theory* $\mathbf{TV}^0$ *has the language* $\mathcal{L}_A^2$ *and axioms those of* $\mathbf{V^0}$ *together with the* Σ_0^B*-String-IND scheme.*

We note that if in the above definition we replaced the Σ_0^B-String-IND scheme by the Σ_i^B-String-IND scheme, $i \ge 1$, we would obtain a theory we might call $\mathbf{TV}^i$, which is the RSUV isomorphism image of Buss's theory $\mathbf{T}_2^i$. However we do not know that $\mathbf{TV}^0$ is isomorphic to $\mathbf{T}_2^0$, since it is not clear that $\mathbf{T}_2^0$ can prove the Σ_0^b-comprehension scheme.

Our proof that all polynomial time functions are Σ_1^B-definable in $\mathbf{TV}^0$ is similar to the proof in [7] that for $i \ge 1$, $\mathbf{T}_2^i$ can Σ_{i+1}^b-define the functions reducible to Σ_i^p. We start by showing that $\mathbf{TV}^0$ proves a suitable recursion scheme.

For each Σ_0^B-formula $\phi(i, X)$ (possibly with other free variables) we define a formula $\phi^{rec}(y, X)$ which says that each bit i of X is defined in terms of the preceding bits of X using ϕ. That is, using the notation $X^{<i}$ for $Chop(X, i)$,

$$\phi^{rec}(y, X) \;\equiv\; \forall i < y(X(i) \leftrightarrow \phi(i, X^{<i}))$$

It is easy to see that $\mathbf{V^0}$ can use induction on y to prove that the condition $\phi^{rec}(y, X)$ uniquely determines bits $X(0), ..., X(y-1)$ of X.

The Σ_0^B-Bit-Recursion scheme is

$$\exists X \phi^{rec}(y, X) \tag{26}$$

where $\phi(i, X)$ is a Σ_0^B formula.

$\mathbf{V^0}$ does not prove the Σ_0^B-Bit-Recursion scheme, since the scheme can be used to Σ_1^B-define Parity(X).

Theorem 3.10 $\mathbf{TV}^0$ *proves the* Σ_0^B*-Bit-Recursion scheme.*

Proof. Let the string reversal function $\mathrm{Rev}(y, X)$ have defining axiom

$$\mathrm{Rev}(y, X)(i) \leftrightarrow i < y \wedge X(y - i - 1)$$

Thus $\mathrm{Rev}(y, X)$ is the reverse of the string $X(0)...X(y-1)$.

We define a formula $\phi^{lessrec}(y, X)$ whose intuitive meaning is $\mathrm{Rev}(y, X) \leq \mathrm{Rev}(y, Y)$, where Y is the unique string satisfying $\phi^{rec}(y, Y) \wedge |Y| \leq y$. This formula asserts that either $X = Y$, or for some j, $X(0), X(1), ..., X(j-1)$ have been computed correctly according to ϕ^{rec} but $X(j)$ is 0 when it should be 1. Thus

$$\phi^{lessrec}(y, X) \;\equiv\; \phi^{rec}(y, X) \vee [|X| \leq y \wedge \exists j < y(\phi^{rec}(j, X) \wedge \neg X(j) \wedge \phi(j, X^{<j}))]$$

Rearranging the String-IND scheme for the formula $\phi^{lessrec}(y, \mathrm{Rev}(y, X))$ we obtain

$$\phi^{lessrec}(y, \mathbf{0}) \wedge \neg\phi^{lessrec}(y, \mathrm{Rev}(y, Y)) \supset \exists X(\phi^{lessrec}(y, \mathrm{Rev}(y, X)) \wedge \neg\phi^{lessrec}(y, \mathrm{Rev}(y, S(X)))$$

Reasoning in $\mathbf{TV}^0$, we may assume $y > 0$, since when $y = 0$ (26) is trivial. Thus $\phi^{lessrec}(y, \mathbf{0})$. Let $Y = \{i \mid i < y\}$ (so Y is the string of ones of length y and $\mathrm{Rev}(y, Y) = Y$). If $\phi^{lessrec}(y, Y)$ then $\phi^{rec}(y, Y)$, so we are done. Hence we may assume $\neg\phi^{lessrec}(y, \mathrm{Rev}(y, Y))$, so we conclude there exists X such that

$$\phi^{lessrec}(y, \mathrm{Rev}(y, X)) \wedge \neg\phi^{lessrec}(y, \mathrm{Rev}(y, S(X)) \tag{27}$$

From this we will argue from the intuitive meaning of $\phi^{lessrec}$ given above that $\phi^{rec}(y, \mathrm{Rev}(y, X))$ holds (as desired). In fact, if X satisfies (27) and Y is the unique string satisfying $\phi^{rec}(y, Y) \wedge |Y| \leq y$ then (by the intuitive meaning) $X \leq \mathrm{Rev}(y, Y)$ but not $S(X) \leq \mathrm{Rev}(y, Y)$, so $X = \mathrm{Rev}(y, Y)$, so $\phi^{rec}(y, \mathrm{Rev}(y, X))$. In fact, it is straightforward to show that $\mathbf{TV^0}$ (and even $\mathbf{V^0}$) proves $\phi^{rec}(y, \mathrm{Rev}(y, X))$ from (27). □

Lemma 3.11 *Suppose* $\mathbf{TV}^0$ Σ_1^B*-defines functions* G *and* H*, and suppose* F *satisfies (22), (23). Then* $\mathbf{TV}^0$ Σ_1^B*-defines* F *and proves formulas equivalent to (22), (23). The same is true when we replace* $\mathbf{TV}^0$ *by* $\mathbf{V^0} + \Sigma_0^B$*-Bit Recursion Scheme.*

Proof. (sketch) The Σ_0^B-Bit Recursion Scheme shows the existence of a string coding the computation of F from G and H. □

Theorem 3.12 **VPV** *is a conservative extension of* $\mathbf{TV}^0$*. The theory* $\mathbf{V^0} + \Sigma_0^B$*-Bit Recursion Scheme is equivalent to* $\mathbf{TV}^0$*.*

Proof. **VPV** proves the Σ_0^B-String-IND scheme for the same reason that **QPV** proves the open-IND scheme (Theorem 3.1). Therefore **VPV** is an extension of $\mathbf{TV}^0$. The extension is conservative by arguing as in the proof of Theorem 2.17, using Lemma 3.11. The last sentence follows for one direction from Theorem 3.10. For the other direction, that $\mathbf{V^0} + \Sigma_0^B$-Bit Recursion Scheme proves Σ_0^B-String-IND follows because the argument that **VPV** is conservative over $\mathbf{TV}^0$ is really an argument that **VPV** is conservative over $\mathbf{V^0} + \Sigma_0^B$-Bit Recursion Scheme, and we argued above that **VPV** proves the Σ_0^B-String-IND scheme. □

Thus $\mathbf{TV}^0$ Σ_1^1-defines exactly the polynomial time functions. It follows from arguments in [18] that $\mathbf{TV}^0$ is equivalent to $\mathbf{V}_1$-**Horn**, and hence $\mathbf{TV}^0$ is finitely axiomatizable. It also follows that $\mathbf{TV}^0$ is RSUV isomorphic to **QPV**, and hence by a result in [4], to the $\forall\Sigma_1^b$ consequences of $\mathbf{S}_2^1$.

4 Theories for other complexity classes

4.1 The theory $\mathbf{VTC}^0$

Nonuniform $\mathbf{TC^0}$ is the class of relations computable by a polynomial-size bounded-depth family of circuits which are allowed threshold gates in addition to the usual Boolean gates. Here the threshold gate $Th_k^n(x_1, ..., x_n)$ is 1 iff at least k of its inputs are 1.

The uniform class $\mathbf{TC^0}$ (our intended meaning) refers to $\mathbf{AC^0}$ (or **FO**) uniformity [28]. From our two-sorted point of view, $\mathbf{TC^0}$ is a class of relations $R(\vec{x}, \vec{X})$, where we pass from languages to relations by representing numbers in unary and finite sets of numbers as bit strings, as usual. This two-sorted $\mathbf{TC^0}$ can be characterized as the closure of $\mathbf{AC^0}$ under Boolean and counting operations [28, 36].

The class $\mathbf{FTC^0}$ of $\mathbf{TC^0}$ functions is defined in the usual way (Definition 2.4) as the class of p-bounded functions whose bit graphs are in $\mathbf{TC^0}$. The class $\mathbf{FTC^0}$ is closed under $\mathbf{AC^0}$ reductions, and both of the functions string multiplication $X \cdot Y$ and $Numones(X)$ are complete for $\mathbf{FTC^0}$ under $\mathbf{AC^0}$ reductions [8]. Here $Numones(X)$ is the number of ones in the string X.

The theory $\mathbf{VTC}^0$ [36, 37] has the language $\mathcal{L}_A^2$ and the axioms of $\mathbf{V^0}$ together the axiom *NUMONES*. ($\mathbf{VTC}^0$ is essentially the same as the theory $(I\Sigma_0^{1,b})^{count}$ proposed by Krajíček [35].) The axiom *NUMONES* states that for each string X, there is a "counting array" Y whose ith row contains an unique number, which is the number of bits in X up to (but not including) position i. Formally, let $\varphi_N(X, Y)$ be the Σ_0^B formula expressing that Y is the counting array for X:

$$\begin{aligned}\varphi_N(X,Y) \quad \equiv \quad & \forall i \le |X| \exists! j \le |X| Y(i,j) \wedge Y(0,0) \wedge \\ & \forall i < |X| \forall j \le |X| [(Y(i,j) \wedge X(i) \supset Y(i+1, j+1)) \wedge (Y(i,j) \wedge \neg X(i) \supset Y(i+1,j))].\end{aligned}$$

Then *NUMONES* is defined as follows.

Definition 4.1 *(NUMONES) The axiom NUMONES is* $\forall X \exists Y \varphi_N(X, Y)$.

Definition 4.2 *($\mathbf{VTC}^0$) The theory* $\mathbf{VTC}^0$ *is* $\mathbf{V^0}$ *together with NUMONES.*

Since $\mathbf{V^0}$ is finitely axiomatizable, it follows that $\mathbf{VTC}^0$ is finitely axiomatizable. Further, $\mathbf{VTC}^0$ is a p-bounded theory, since the quantifier $\exists Y$ in *NUMONES* can be provably bounded by $1 + \langle |X|, |X| \rangle$, using Σ_0^B-COMP.

Now we introduce a universal theory $\overline{\mathbf{VTC}^0}$ which is a conservative extension of $\mathbf{VTC}^0$. We start by introducing the number function $numones(i, X)$ whose value is the number of ones among the bits $X(0), ..., X(i-1)$. The defining axioms for $numones$ are

$$\begin{aligned}&numones(0, X) = 0\\ &X(i) \supset numones(i+1, X) = numones(i, X) + 1\\ &\neg X(i) \supset numones(i+1, X) = numones(i, X)\end{aligned} \tag{28}$$

Let $\mathbf{VTC}^0(numones)$ be the theory resulting by adding the function $numones$ and its defining axioms (28) to $\mathbf{VTC}^0$.

Lemma 4.3 $\mathbf{VTC}^0(numones)$ *is a conservative extension of* $\mathbf{VTC}^0$*, and* $\mathbf{VTC}^0(numones)$ *proves the* $\Sigma_0^B(numones)$*-COMP scheme.*

Proof. To prove conservativity, we simply note that by the axiom *NUMONES*, any model of $\mathbf{VTC}^0$ can be expanded to a model of $\mathbf{VTC}^0(numones)$.

To prove the second sentence, we must show

$$\mathbf{VTC}^0(numones) \vdash \exists Z \le y \forall i < y[Z(i) \leftrightarrow \phi(i, \vec{x}, \vec{X})] \tag{29}$$

for each $\Sigma_0^B(numones)$-formula ϕ. We prove this by induction on the logical depth of ϕ. The induction step follows easily by Σ_0^B-COMP. The base cases are when ϕ is atomic; that is one of the three forms $t = u, t \le u, X(t)$ for terms t, u over $\mathcal{L}_A^2(numones)$. All three subcases follow from the lemma below. We illustrate for the subcase $\phi(i) \equiv X(t(i))$. By the lemma,

$$\mathbf{VTC}^0(numones) \vdash \exists W \forall i < y \forall k \le p(i)[W(i,k) \leftrightarrow k = t(i)] \tag{30}$$

where $p(i)$ is a term of $\mathcal{L}_A^2$ bounding $t(i)$. But by Σ_0^B-COMP

$$\mathbf{V^0} \vdash \exists Z \le y \forall i < y[Z(i) \leftrightarrow \exists k \le p(i)(W(i,k) \wedge X(k))] \tag{31}$$

Now (29) follows from (30) and (31) when ϕ is $X(t(i))$. $\square$

Lemma 4.4 *For every term* $t = t(\vec{x}, \vec{X})$ *of* $\mathcal{L}_A^2(numones)$

$$\mathbf{VTC}^0(numones) \vdash \exists Z \forall \vec{x}, k < y[Z(\vec{x}, k) \leftrightarrow k = t(\vec{x}, \vec{X})]$$

Proof. This is a straightforward induction on the depth of t, using the axiom *NUMONES*. $\square$

The language $\mathcal{L}_{\mathbf{FTC^0}}$ of $\overline{\mathbf{VTC}^0}$ extends $\mathcal{L}_{\mathbf{FAC^0}}$ and has functions symbols for all functions in $\mathbf{FTC^0}$. We give an inductive definition of the functions in $\mathcal{L}_{\mathbf{FTC^0}}$, in the same style as given for $\mathcal{L}_{\mathbf{FAC^0}}$ in Section 2.6.

(a$''$) $\mathcal{L}_{\mathbf{FTC^0}}$ includes $\mathcal{L}_A^2 \cup \{pd, numones\}$.

(b$''$), (c$''$) are the same as (b), (c) in Section 2.6 except $\mathcal{L}_{\mathbf{FAC^0}}$ is replaced by $\mathcal{L}_{\mathbf{FTC^0}}$.

Definition 4.5 $\overline{\mathbf{VTC}^0}$ *is the theory whose language is* $\mathcal{L}_{\mathbf{FTC^0}}$ *and whose axioms are the universal closures of* **B1**,...,**B13**, **B14$'$**,**B14$''$**,**L1**,**L2**,**SE***, the defining axioms (28) for numones, and the defining axiom (12) for each function* $F_{\alpha,t}$ *and defining axioms (13,14,15) for each function* $f_{\alpha,t}$.

Theorem 4.6 $\overline{\mathbf{VTC}^0}$ *is a universal theory which is a conservative extension of* $\mathbf{VTC}^0$. *The function symbols of* $\mathcal{L}_{\mathbf{FTC^0}}$ *represent precisely the functions in* $\mathbf{FTC^0}$.

Proof. The last sentence follows from the fact that $\mathbf{FTC^0}$ is the closure of the function *numones* under $\mathbf{AC^0}$ reductions.

To show that $\overline{\mathbf{VTC}^0}$ extends $\mathbf{VTC}^0$ it suffices to show that $\overline{\mathbf{VTC}^0}$ proves *NUMONES*. This is because $\varphi_N(X,Y)$ defines the graph $Y = Numones(X)$ of a string function whose bit graph is easily defined by a Σ_0^B-(*numones*) formula. Thus $Numones(X)$ is represented by a function $F_{\alpha,t}$ in $\mathcal{L}_{\mathbf{FTC^0}}$, and the axiom *NUMONES* follows.

That $\overline{\mathbf{VTC}^0}$ is conservative over $\mathbf{VTC}^0$ follows from Lemma 4.3 and the following general lemma. □

Lemma 4.7 *Let T be a theory over a language $\mathcal{L}$, and suppose that T extends* $\mathbf{V^0}$ *and proves the* $\Sigma_0^B(\mathcal{L})$*-COMP scheme. Let F be a new function with bit-defining axiom*

$$AX(F): \qquad F(\vec{x},\vec{X})(i) \leftrightarrow i < t \wedge \phi(i,\vec{x},\vec{X})$$

where ϕ is a $\Sigma_0^B(\mathcal{L})$ formula. Let T' be obtained from T by adding F and its defining axiom $AX(F)$, so the language of T' is $\mathcal{L}' = \mathcal{L} \cup \{F\}$. Then T' is a conservative extension of T, and T' proves the $\Sigma_0^B(\mathcal{L}')$*-COMP scheme.*

Proof. Since T proves $\Sigma_0^B(\mathcal{L})$-COMP scheme, it follows that this theory $\Sigma_1^B(\mathcal{L})$-defines F, so T' is conservative over T.

We show that T' proves the comprehension axiom for each $\Sigma_0^B(\mathcal{L}')$-formula ψ by induction on the logical depth of ψ. The induction step follows easily from the $\Sigma_0^B(\mathcal{L})$-COMP scheme. For the base case (ψ is atomic) we use Lemma 4.8 below.

For example, suppose that $\psi(i)$ is $R(i,t(i),C)$ where R is a predicate symbol, $t(i)$ is a number term and C is a string term not involving i. (The case in which C involves i can also be handled by the lemma.) We are to show

$$T' \vdash \exists Z \forall i < y[Z(i) \leftrightarrow R(i,t(i),C)] \tag{32}$$

This follows from the following two consequences of the lemma, using $\Sigma_0^B(\mathcal{L})$-COMP:

$$T' \vdash \exists W \forall j < q[W(j) \leftrightarrow C(j)]$$

where q is a $\mathcal{L}_A^2$ term giving an upper bound on $|C|$.

$$T' \vdash \exists U \forall i < y \forall k < p(i)[U(i,k) \leftrightarrow k = t(i)]$$

where $p(i)$ a $\mathcal{L}_A^2$ term giving an upper bound on $t(i)$. □

Lemma 4.8 *(a) For each string term $C(\vec{x},\vec{X})$ in $\mathcal{L}'$*

$$T' \vdash \exists W \forall \vec{x}, i < y[W(\vec{x},i) \leftrightarrow C(\vec{x},\vec{X})(i)]$$

(b) For each number term $t(\vec{x},\vec{X})$ in $\mathcal{L}'$

$$T' \vdash \exists Z \forall \vec{x}, k < y[Z(\vec{x},k) \leftrightarrow k = t(\vec{x},\vec{X})]$$

Proof. Parts (a) and (b) are proved together by induction on d, where d bounds the depth of the terms C and t. $\square$

It follows from Theorem 2.19 (witnessing) that Σ_1^1 theorems of $\overline{\mathbf{VTC}^0}$ are witnessed by functions in $\mathbf{FTC^0}$. The next result follows from this and Theorem 4.6 and the fact that every $\Sigma_0^B(\mathcal{L}_{\mathbf{FTC^0}})$-formula is provably equivalent to an open formula (see Lemma 2.16).

Theorem 4.9 *A function is Σ_1^1-definable in $\mathbf{VTC}^0$ iff is is in $\mathbf{FTC^0}$.*

Since binary multiplication $X \cdot Y$ is a $\mathbf{TC^0}$ function, it is definable in $\mathbf{VTC}^0$. Nguyen [36] proves that $\mathbf{VTC}^0$ proves $X \cdot Y = Y \cdot X$, and uses this in proving the interesting result that $\mathbf{VTC}^0$ is RSUV isomorphic to the different-looking single-sorted theory Δ_1^b-$\mathbf{CR}$ of Johannsen and Pollett [29].

Earlier we pointed out that the pigeonhole principle $PHP(n, X)$ (10) is not provable in $\mathbf{V^0}$, intuitively because counting functions are not definable in $\mathbf{V^0}$. As one might expect, $PHP(n, X)$ is provable in $\mathbf{VTC}^0$ [36]. As a consequence of this and results on propositional translations of $\mathbf{VTC}^0$, it follows that the propositional pigeonhole principle PHP_n^{n+1} has polynomial size $\mathbf{TC^0}$-Frege proofs (see Section 5.3).

It would be worthwhile to see whether other counting arguments can be formalized in $\mathbf{VTC}^0$. For example, Buss [11] showed that tautologies related to the game of Hex have polynomial-size $\mathbf{TC^0}$-Frege proofs, and presumably his interesting arguments can be formalized in $\mathbf{VTC}^0$.

4.2 A theory for $\mathbf{AC^0}(2)$

Nonuniform $\mathbf{AC^0}(2)$ is the class of relations computable by a polynomial-size bounded-depth family of circuits which are allowed unbounded fanin parity gates $\oplus(x_1, ..., x_n)$ in addition to the usual Boolean gates. The uniform class refers to $\mathbf{FO}$ uniformity [28].

Let $Parity(X)$ be the relation which holds iff X has an odd number of 1's. Then our class $\mathbf{AC^0}(2)$ can be defined as the class of all relations $R(\vec{x}, \vec{X})$ which are $\mathbf{AC^0}$ reducible to $Parity$.

We define a theory $\mathbf{V^0}(2)$ by adding an axiom to $\mathbf{V^0}$ stating the existence of a parity function. Specifically, we define a Σ_0^B formula $par(X, Y)$ giving the graph of the function as follows:

$$par(X, Y) \ \equiv \ |Y| \leq |X| + 1 \wedge \neg Y(0) \wedge \forall i < |X|(Y(i+1) \leftrightarrow Y(i) \oplus X(i))$$

Definition 4.10 *$\mathbf{V^0}(2)$ is the theory which has the same language $\mathcal{L}_A^2$ as $\mathbf{V^0}$, and extends $\mathbf{V^0}$ by adding the single axiom*

$$\exists Y\, par(X, Y)$$

The theory of $\mathbf{V^0}(2)$ can be developed in a manner analogous to that of $\mathbf{VTC}^0$. In particular, we have

Theorem 4.11 *A function is Σ_1^1-definable in $\mathbf{V^0}(2)$ iff it is in $\mathbf{FAC^0}(2)$.*

4.3 The theory $\mathbf{VNC}^1$

A problem in nonuniform $\mathbf{NC}^1$ is given by a polynomial-size log-depth family of Boolean circuits, or equivalently by a polynomial-size family of propositional formulas. Uniform $\mathbf{NC}^1$ (which we use here) means **Alogtime**. Thus a two-sorted relation $R(\vec{x}, \vec{X})$ is in $\mathbf{NC}^1$ iff it is recognized by some alternating Turing machine in time $O(\log n)$. Buss [5] proved that the Boolean formula value problem is complete for **Alogtime**.

Our two-sorted theory $\mathbf{VNC}^1$ [20] is inspired by Arai's [3] single-sorted theory **AID**. We define $\mathbf{VNC}^1$ by adding a tree recursion axiom scheme Σ_0^B-*TreeRec* to $\mathbf{V^0}$.

The Σ_0^B-*TreeRec* scheme (which uses the idea of a heap data structure) is

$$\exists Z \leq 2a \forall i < a[(Z(i+a) \leftrightarrow \psi(i)) \wedge 0 < i \supset (Z(i) \leftrightarrow \phi(i)[Z(2i), Z(2i+1)])] \tag{33}$$

where $\phi(i)[p,q]$ and $\psi(i)$ are Σ_0^B formulas (which do not contain Z but may contain other parameters) and ϕ contains atoms p, q which are replaced in the axiom by $Z(2i), Z(2i+1)$.

The idea is that the vector Z assigns truth values to the nodes of a binary tree, where the nodes are indexed by the variable $i, 0 < i \leq 2a-1$. The leaves of the tree are indexed by any i such that $a \leq i \leq 2a-1$ and leaf number i is assigned value $\psi(i)$. The internal nodes of the tree are indexed by any i such that $1 \leq i \leq a-1$, and the value $Z(i)$ of node i is determined by the values $Z(2i), Z(2i+1)$ of its two children by the formula ϕ. The root of the tree is indexed by $i = 1$, so $Z(1)$ is the output of the recursion.

We can define $\mathbf{NC}^1$ relations in $\mathbf{VNC}^1$ as follows. For Σ_0^B formulas $\phi(i, \vec{x}, \vec{X})[p,q]$ and $\psi(i, \vec{x}, \vec{X})$ in the Σ_0^B-*TreeRec* scheme (33) we define the Σ_0^B formula $B^{\phi,\psi}(a, \vec{x}, \vec{X}, Z)$ to be the part of (33) which comes after $\exists Z \leq 2a$. That is,

$$B^{\phi,\psi}(a, \vec{x}, \vec{X}, Z) \equiv \forall i < a[(Z(i+a) \leftrightarrow \psi(i)) \wedge 0 < i \supset (Z(i) \leftrightarrow \phi(i)[Z(2i), Z(2i+1)])]$$

Every formula $B^{\phi,\psi}$ defines a relation $R^{\phi,\psi}$ (computed by the recursion scheme (33)) with defining axiom

$$R^{\phi,\psi}(a, i, \vec{x}, \vec{X}) \leftrightarrow \exists Z \leq 2a(B^{\phi,\psi}(a, \vec{x}, \vec{X}, Z) \wedge Z(i)) \tag{34}$$

Lemma 4.12 *The relation $R^{\phi,\psi}$ is in $\mathbf{NC}^1$, for each pair $\phi[p,q], \psi$ of Σ_0^B formulas.*

Proof. We start by observing that each Σ_0^B formula represents an $\mathbf{AC^0}$ relation, which is therefore in **Alogtime**. To prove the lemma, it suffices to show there exists an indexed alternating Turing machine M with inputs $(a, i, \vec{x}, \vec{X})$ (where number inputs are presented in unary notation) which computes $R^{\phi,\psi}$ in time $O(\log n)$, where n is the length of the input. The machine M recursively guesses $Z(j)$ and verifies its guess. Initially M guesses $Z(i)$ is true. In general, M verifies its guess for $Z(j)$ as follows: First it guesses whether $j < a$ or $j \geq a$. If the guess is $j \geq a$, then it verifies the guess, and verifies $Z(j) \leftrightarrow \psi(j, \vec{x}, \vec{X})$, all in time $O(\log n)$. If the guess is $j < a$ it branches universally, verifying the guess on one branch and guessing $Z(2j), Z(2j+1)$ on the other branch. After the second branch it next does a three-way universal branch: (i) verify $Z(j) \leftrightarrow \phi(j, \vec{x}, \vec{X})[Z(2j), Z(2j+1)]$, (ii) verify $Z(2j)$ recursively, and (iii) verify $Z(2j+1)$ recursively.

Note that the depth of the recursion is proportional to the depth of the tree recursion defined by (33), which is $O(\log a) = O(\log n)$. $\square$

We now expand the language $\mathcal{L}_A^2$ to $\mathcal{L}_{TreeRec}$ by putting in a predicate symbol $R^{\phi,\psi}$ for each relation $R^{\phi,\psi}$ defined in (34). Then $\Sigma_0^B(\mathcal{L}_{TreeRec})$ denotes the class of formulas in this language with no string quantifiers, and all number quantifiers bounded.

Lemma 4.13 *The class of $\Sigma_0^B(\mathcal{L}_{TreeRec})$ formulas represents precisely the* $\mathbf{NC}^1$ *relations.*

Proof. Every such formula represents an $\mathbf{NC}^1$ relation, by the previous lemma, and the easy fact that the $\mathbf{NC}^1$ relations are closed under bounded number quantification and the Boolean operations.

The converse involves showing that our tree recursion scheme can simulate alternating Turing machines. The idea is that the recursion tree represents the computation of the machine. This has been worked out in detail in the context of the single-sorted theory **AID** [3]. □

Let $\mathbf{FNC}^1$ be the function class associated with $\mathbf{NC}^1$ (see Definition 2.4).

The next two theorems are proved in [20].

Theorem 4.14 *The Σ_1^1-definable functions in* $\mathbf{VNC}^1$ *are precisely those in* $\mathbf{FNC}^1$.

In the next result, Σ_0^b-CA [3] refers to the comprehension scheme for sharply bounded formulas over the base language $\mathcal{L}_{BA}$.

Theorem 4.15 $\mathbf{VNC}^1$ *is RSUV isomorphic to* **AID** + Σ_0^b*-CA.*

4.4 A theory for L

The nonuniform class **L** (log space) consists of problems solvable by a polynomial-size family of branching programs. Here **L** refers to the two-sorted uniform class of all relations $R(\vec{x},\vec{X})$ solvable in space $O(\log n)$ on a Turing machine. We use **FL** to denote the corresponding function class, given by Definition 2.4.

The relevant two-sorted theory is **VL**, which is our name for the theory Σ_0^B-rec, due to Zambella [49]. This theory has the same language $\mathcal{L}_A^2$ as $\mathbf{V^0}$, and extends $\mathbf{V^0}$ by the following recursion scheme, where ϕ is a Σ_0^B-formula not involving Z:

$$\forall x \le a \exists y \le a\, \phi(x,y) \supset \exists Z \forall w < b\, \phi(Z(w), Z(w+1)) \tag{35}$$

where $Z(x)$ is the value at x of the function coded by the string Z using some natural coding (see for example the axiom *NUMONES* in Definition 4.1).

It is not hard to see that a log space Turing machine can witness the quantifier $\exists Z$ above by computing a sequence $Z(0), Z(1), ..., Z(b-1)$ using the recurrence

$$\begin{aligned} Z(0) &= 0 \\ Z(w+1) &= \min y < a\, \phi(Z(w), y) \end{aligned}$$

It follows by a standard witnessing argument that every Σ_1^1-theorem of the theory **VL** can be witnessed by functions in **FL**.

Conversely, Zambella [49] shows that every function in **FL** is Σ_1^1-definable in **VL**.

A propositional proof system for **L** is presented in the slides [16], based on branching programs and the Pudlak-Buss liar game [40].

4.5 A theory for NL

NL is the class of problems solvable by a nondeterministic Turing machine in space $O(\log n)$. As usual, we consider **NL** as a class of two-sorted relations $R(\vec{x}, \vec{X})$, and the corresponding function class **FNL** is given by Definition 2.4. It is not at all obvious that **NL** is closed under complementation, and before this was proved [27, 46] it was not known whether **FNL** was closed under composition. However we now know that **FNL** is a robust class, and is closed under $\mathbf{AC^0}$ reductions [28, 19].

Our two-sorted theory **V-Krom** is the analog for **NL** of $\mathbf{V}_1$-**Horn** for **P** (Section 3.4). It is based on Grädel's second-order descriptive characterization of **NL**, which in turn is based on the fact that the satisfiability problem for propositional 2CNF (conjunctive normal form formulas with two literals per clause) is complete for co**NL** (and hence complete for **NL**).

A Σ_1-Krom formula is one of the form

$$\exists \vec{P} \forall \vec{x} < \vec{t} \phi(\vec{P}, \vec{x}, \vec{y}, \vec{X})$$

where ϕ is a CNF formula of $\mathcal{L}_A^2$ in which each clause has at most two occurrences from $\vec{P}$, and each occurrence must be as a literal of the form $P_i(t)$ or $\neg P_i(t)$.

Lemma 4.16 *The Σ_1-Krom formulas represent precisely the* **NL** *relations.*

Definition 4.17 *The theory* **V-Krom** *has the language and axioms of* $\mathbf{V^0}$, *together with the comprehension scheme for all Σ_1-Krom formulas.*

In [19] it is proved that the Σ_1^1-definable functions in **V-Krom** are precisely those in **FNL**. This proof is difficult, since it involves showing that the Immerman-Szelepcsinyi theorem can be formalized in **V-Krom**.

4.6 A theory for NC

For each $k \geq 1$ a problem in nonuniform $\mathbf{NC}^k$ is presented by a polynomial-size family of circuits of depth $O(\log^k n)$. Then $\mathbf{NC} = \bigcup_k \mathbf{NC}^k$. Our **NC** (uniform **NC**) is the class of two-sorted relations $R(\vec{x}, \vec{X})$ recognized by some alternating Turing machine in time $O(\log^k n)$ and space $O(\log n)$. The function class **FNC** is defined from **NC** in the usual way (Definition 2.4).

Several single-sorted theories corresponding to **NC** have been proposed, and Takeuti [47] proved that the two-sorted theory $\mathbf{U}_1^1(BD)$ is isomorphic to one of them. Here we present the two-sorted theory $\mathbf{U}^1$, which our version of $\mathbf{U}_1^1(BD)$.

Definition 4.18 *The theory $\mathbf{U}^1$ has the language and axioms of* $\mathbf{V^0}$, *together with the Σ_1^B-LIND scheme*

$$[\phi(0) \wedge \forall x(\phi(x) \supset \phi(x+1))] \supset \phi(|z|) \tag{36}$$

where ϕ is a Σ_1^B formula.

Here $|z|$ is the binary length of z. The function $|z|$ is not in the vocabulary $\mathcal{L}_A^2$ of $\mathbf{V^0}$, but it can be Σ^b-defined in $\mathbf{I\Delta_0}$ and its properties proved [9, 17] and hence it can be Σ_0^B-defined and its properties proved in $\mathbf{V^0}$. Thus we take (36) to denote the equivalent formula in the language $\mathcal{L}_A^2$.

From results in [47, 12] it follows that the Σ_1^1-definable functions in $\mathbf{U}^1$ are those in **FNC**.

For each $k \geq 0$ Clote and Takeuti define a first-order theory TNC^k which corresponds to the complexity class $\mathbf{NC}^{k+1}$. One can define a propositional proof system $\mathbf{NC}^{k+1}$-Frege by restricting the depth of nesting of extension definitions in an extended Frege system to be $\log^k n$. So far no one has given an explicit translation between a theory for $\mathbf{NC}^{k+1}$ and a propositional proof system, although this should not be difficult.

5 Translations into the quantified propositional calculus

Propositional translations of theories of arithmetic (see [31] for a good general reference) goes back at least to Cook [14], who outlined a method of translating theorems of the equational system **PV** into polynomial-size families of propositional proofs in the system ER (extended resolution). Later Paris and Wilkie [39] gave a simpler and more elegant translation of bounded theorems of the relativized theory $\mathbf{I\Delta_0}(R)$ into polynomial-size families of bounded-depth Frege proofs. The bounded theorems of $\mathbf{I\Delta_0}(R)$ are closely related to the Σ_0^B theorems of $\mathbf{V^0}$ (Section 2.4), and the translations we describe below are a generalization of the Paris-Wilkie method.

Krajíček and Pudlak [32] introduced the system G for the quantified propositional calculus (QPC), together with the fragments G_i and G_i^*. For $i \geq 1$ they described translations of Σ_i^b theorems of Buss's theories $\mathbf{T}_2^i$ and $\mathbf{S}_2^i$ to polynomial-size families of proofs in G_i and G_i^*, respectively. These translations can be regarded as a generalization of Cook's translation of **PV**.

Here we use the elegant method of Paris and Wilkie to give a simple translation of any bounded formula in our two-sorted language $\mathcal{L}_A^2$ to a polynomial-size family of formulas of QPC. Further we adopt a modification of the QPC systems G_i and G_i^* proposed by Morioka (see [20]) and suggest that the Krajíček -Pudlak translations of $\mathbf{T}_2^i$ and $\mathbf{S}_2^i$ can be described more simply and naturally as Paris-Wilkie-style translations from $\mathbf{TV}^i$ and $\mathbf{V}^i$, their two-sorted images, to the modified G_i and G_i^*.

5.1 The systems G, G_i, and G_i^*

The system G for QPC is a generalization of Gentzen's sequent system PK for the propositional calculus (see [31, 10, 17]). The logical symbols include the truth constants $\mathtt{T}$ and $\mathtt{F}$, the propositional connectives $\wedge, \vee, \neg$, and the quantifiers $\forall, \exists$. There are two sets of propositional variables: $p, q, r, ...$ for free variables and $x, y, z, ...$ for bound variables. Formulas are defined in the usual way from these symbols, and sequents have the form $\Gamma \to \Delta$, where Γ and Δ are finite sequences of formulas.

The axioms (initial sequents) include $A \to A$ for any quantifier-free formula A, and $\mathtt{F} \to$ and $\to \mathtt{T}$. The rules of inference include the usual structural rules (weakening, contraction, exchange), the cut rule, and propositional rules for introducing each of the connectives $\wedge, \vee, \neg$ into each side of a sequent.

In addition, there are four quantifier rules as follows:

$$\exists\text{-left} : \frac{A(b), \Gamma \to \Delta}{\exists x A(x), \Gamma \to \Delta} \qquad \exists\text{-right} : \frac{\Gamma \to \Delta, A(B)}{\Gamma \to \Delta, \exists x A(x)}$$
$$\forall\text{-left} : \frac{A(B), \Gamma \to \Delta}{\forall x A(x), \Gamma \to \Delta} \qquad \forall\text{-right} : \frac{\Gamma \to \Delta, A(b)}{\Gamma \to \Delta, \forall x A(x)}$$

where b is an eigenvariable not occurring in the bottom sequent and B is any proper formula. We call B the *target* of the corresponding $\exists$-right or $\forall$-left step.

We put in the restriction from [20], not included in [32], that the the target of every $\forall$-left and $\exists$-right be quantifier-free.

Both Σ_0^q and Π_0^q denote the set of quantifier-free formulas. For $i \geq 1$, Σ_i^q and Π_i^q are the sets of QPC formulas A such that A has a prenex form with at most $i-1$ quantifier alternations, starting with a block of (zero or more) $\exists$'s and a block of (zero or more) $\forall$'s, respectively.

Definition 5.1 *For $i \geq 0$, G_i is G with cuts restricted to $\Sigma_i^q \cup \Pi_i^q$-formulas. G_i^* is the tree-like version of G_i.*

This definition differs from that in [32] in that the latter restricts all formulas in a G_i proof or a G_i^* proof to be in $\Sigma_i^q \cup \Pi_i^q$. Our definition allows systems G_i and G_i^* to prove arbitrary valid formulas of QPC, and it turns out the new definitions give theories p-equivalent to the originals when proving formulas in $\Sigma_i^q \cup \Pi_i^q$.

Note that the theory G_0 does not exist as a QPC theory in [32]. However, Theorem 5.2 below uses our modified definition of G_0 to state that bounded theorems of $\mathbf{VNC}^1$ translate to polynomial size families of quantified G_0 theorems.

5.2 Translations from $\mathcal{L}_A^2$ to QPC

Here we describe the translation from [20], which is a modified version of that in [17].

We translate each bounded predicate calculus formula $\phi(\vec{X})$ over $\mathcal{L}_A^2$ to a polynomial size family $\|\phi(\vec{X})\|[\vec{n}]$ of formulas of the quantified propositional calculus. For each string variable X we associate the propositional variables $p_0^X, p_1^X, \ldots$ where p_i^X is intended to mean $X(i)$. We assume that $\phi(\vec{X})$ has no free number variables, since we will replace all such variables by number constants. The translation has the property that for each $n \in \mathbb{N}$, $\|\phi(X)\|[n]$ is valid iff the formula $\forall X(|X| = n \supset \phi(X))$ is true in the standard model. More generally, there is a one-one correspondence between truth assignments satisfying $\|\phi(\vec{X})\|[\vec{n}]$ and tuples of strings $\vec{X}$, with $|X_i| = n_i$, satisfying $\phi(\vec{X})$.

We use the notation $val(t)$ for the numerical value of a term t, where t may have numerical constants substituted for variables.

The first step in defining $\|\phi(\vec{X})\|[\vec{n}]$ is to replace every atomic formula of the form $X = Y$ by its Σ_0^B definition, given by the RHS of the extensionality axiom SE. After this is done, we define

$\|\phi(\vec{X})\|[\vec{n}]$ by structural induction on the resulting formula $\phi(\vec{X})$. The base case is when $\phi(\vec{X})$ is atomic. If $\phi(\vec{X})$ is $\mathtt{T}$ or $\mathtt{F}$ then $\|\phi(\vec{X})\|[\vec{n}] = \phi(\vec{X})$. If $\phi(\vec{X})$ is $t(|\vec{X}|) = u(|\vec{X}|)$, then $\|\phi(\vec{X})\|[\vec{n}] = \mathtt{T}$ if $val(t(\vec{n})) = val(u(\vec{n}))$ and $\|\phi(\vec{X})\|[\vec{n}] = \mathtt{F}$ otherwise. Similarly if $\phi(\vec{X})$ is $t(|\vec{X}|) \leq (|\vec{X}|)$.

If $\phi(\vec{X})$ is $X_i(t(|\vec{X}|))$, then we set $j = val(t(\vec{n})$ and

$$\|\phi(\vec{X})\|[\vec{n}] = \begin{cases} p_j^{X_i} & \text{if } j < n_i - 1 \\ \mathtt{T} & \text{if } j = n_i - 1 \\ \mathtt{F} & \text{if } j > n_i - 1 \end{cases}$$

For the induction step, $\phi(\vec{X})$ is built from smaller formulas using a propositional connective $\wedge, \vee, \neg$, or a bounded quantifier. For $\wedge, \vee, \neg$ we make the obvious definition; for example

$$\|\psi(\vec{X}) \wedge \eta(\vec{X})\|[\vec{n}] = (\|\psi(\vec{X})\|[\vec{n}] \wedge \|\eta(\vec{X})\|[\vec{n}])$$

For the case of bounded number quantifiers, we define

$$\begin{aligned} \|\exists y \leq t(|\vec{X}|)\psi(y, \vec{X})\|[\vec{n}] &= \bigvee_{i=0}^{m} \|\psi(i, \vec{X})\|[\vec{n}] \\ \|\forall y \leq t(|\vec{X}|)\psi(y, \vec{X})\|[\vec{n}] &= \bigwedge_{i=0}^{m} \|\psi(i, \vec{X})\|[\vec{n}] \end{aligned}$$

where $m = val(t(\vec{n}))$.

Finally, for the case of bounded string quantifiers, we define

$$\begin{aligned} \|\exists Y \leq t(|\vec{X}|)\psi(Y, \vec{X})\|[\vec{n}] = \exists p_0^Y ... \exists p_{m-2}^Y \textstyle\bigvee_{i=0}^m \|\psi(Y, \vec{X})\|[i, \vec{n}] \\ \|\forall Y \leq t(|\vec{X}|)\psi(Y, \vec{X})\|[\vec{n}] = \forall p_0^Y ... \forall p_{m-2}^Y \textstyle\bigwedge_{i=0}^m \|\psi(Y, \vec{X})\|[i, \vec{n}] \end{aligned}$$

where again $m = val(t(\vec{n}))$. (To meet our free-bound variable convention, each quantified variable p_i^Y above should be replaced by a "bound" variable x_i^Y.)

This completes the definition of the translation $\|\phi(\vec{X})\|[\vec{n}]$ of $\phi(\vec{X})$. Notice that Σ_i^B formulas translate to families of Σ_i^q formulas.

We handle free number variables in ϕ by substituting numerical constants (numerals) for them. Given any formula $\phi(\vec{x}, \vec{X})$ over the language $\mathcal{L}_A^2$ there is a polynomial $p(\vec{x}, \vec{y})$ such that the QPC formula $\|\phi(\vec{r}, \vec{X})\|[\vec{n}]$ is bounded in size by $p(\vec{r}, \vec{n})$. Further, if $\phi(\vec{x}, \vec{X})$ is Σ_0^B, then the $\wedge$ - $\vee$ alternation depth of $\|\phi(\vec{r}, \vec{X})\|[\vec{n}]$ is bounded, independent of $\vec{r}, \vec{n}$.

5.3 Connecting theories and propositional systems

This translation allows us to restate a number of results from the literature. For example, the Paris-Wilkie translation from $\mathbf{I\Delta_0}(R)$ can be generalized to say that any Σ_0^B-theorem $\phi(\vec{x}, \vec{X})$ of $\mathbf{V^0}$ translates into a family $\|\phi(\vec{r}, \vec{X})\|[\vec{n}]$ of propositional formulas with polynomial-size bounded-depth Frege proofs. From this we can conclude that the pigeonhole principle $PHP(n, X)$ (10) is not a theorem $\mathbf{V^0}$, since it is known [2] that its propositional translations PHP_n^{n+1} do not have polynomial-size bounded-depth Frege proofs.

On the other hand, a modification of our propositional translations allows Σ_0^B-theorems of the theory $\mathbf{VTC}^0$ to be translated into polynomial-size $\mathbf{TC^0}$-Frege proofs [37] (which allow bounded-depth formulas with threshold gates). It is not hard to formalize a proof of $PHP(n, X)$ in $\mathbf{VTC}^0$ [31], and as a result we obtain polynomial-size $\mathbf{TC^0}$-Frege proofs for PHP_n^{n+1}. Buss [6] gave polynomial-size Frege proofs for PHP_n^{n+1} before the system $\mathbf{TC^0}$-Frege was defined. The Frege proofs require introduction of counting formulas, together with short Frege proofs of their properties. Buss's techniques also show that Frege systems can efficiently simulate $\mathbf{TC^0}$-Frege systems.

Adapting the arguments in [32], we can show for $i \geq 1$ that any bounded theorem of $\mathbf{TV}^i$ or $\mathbf{V}^i$ translates into a family of QPC formulas with polynomial-size G_i or G_i^* proofs, respectively.

For the case $i = 0$ we state a result from [20].

Theorem 5.2 *If $\phi(\vec{x}, \vec{X})$ is a bounded theorem of $\mathbf{VNC}^1$, then the family $\|\phi(\vec{r}, \vec{X})\|[\vec{n}]$ has G_0^* proofs of size polynomial in $\vec{r}, \vec{n}$. In particular, if ϕ is a Σ_0^B-formula, then its translations have polynomial-size Frege proofs.*

5.4 Polynomial space

The validity problem for formulas in QPC is complete for **PSPACE** (polynomial space). Hence it seems natural that a theory for **PSPACE** should translate into all of G, as opposed to its levels G_i.

Dowd [23] introduced the single-sorted equational theory **PSA** for **PSPACE**. This theory is similar to **PV** (Section 3.1), except new functions are introduced by bounded recursion instead of bounded recursion on notation. [23] gives a translation of **PSA** into a QPC system equivalent to G. Since function symbols represent arbitrary **PSPACE** functions, the translation of a single equation of **PSA** is to a family of QPC formulas whose quantifier complexity increases with the input length.

Buss [4] introduced the two-sorted theory $\mathbf{U}_2^1$ for **PSPACE**, where $\mathbf{U}_2^1$ is essentially $\mathbf{U}^1$ with the function $x\#y$. However, the PSPACE functions in $\mathbf{U}_2^1$ are functions on the number sort, as opposed to the string sort, and strings are used to represent computations rather than inputs. Krajíček and Takeuti [34] translate bounded "first-order" (number sort) theorems of $\mathbf{U}_2^1$ into polynomial-size families of G theorems. Each formula in $\Sigma_0^B(\#)$ with no string variable represents concepts in some fixed level of the polynomial hierarchy (as opposed to arbitrary **PSPACE**), and translates into a QPC family with fixed quantifier complexity. However the G proofs of members of the family involve formulas whose quantifier complexity grows with the length of the input. The translation is in the style of the **PV** translation, rather than the simpler string translation $\|\phi(\vec{X})\|[\vec{n}]$ that we define here.

Skelley [43] introduced a three-sorted theory $\mathbf{W}_1^1$ for **PSPACE** and gives a simple string translation of it into a system BPLK which is equivalent to G for propositional formulas.

6 Conclusion

We have associated a two-sorted theory with each of the complexity classes

$$\mathbf{AC^0} \subset \mathbf{AC^0}(2) \subset \mathbf{TC^0} \subseteq \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC} \subseteq \mathbf{P} \tag{37}$$

Each of the theories has the same underlying language $\mathcal{L}_A^2$, and we have argued that each theory is "minimal" with respect to the condition of being able to formalize proofs which can use arbitrary concepts from the associated complexity class. In most cases a corresponding propositional proof system has been defined (either here or elsewhere), although apparently not yet in the case of **NL**, and there is probably a formula-based system for **L** which would be preferably to the one based on branching programs presented in [16].

The standard used in this paper for associating a theory with a complexity class is that the functions Σ_1^1-definable in the theory are precisely those in the class. This standard applies to the theories we associate with each of the classes in (37), although it does not apply to the theories $\mathbf{V}^i$ associated with the polynomial hierarchy for $i \geq 2$. Thus the functions in $\square_i^p$ (those reducible to Σ_{i-1}^p) are those Σ_i^B-definable in $\mathbf{V}^i$, as opposed to those Σ_1^1-definable or Σ_1^B-definable in $\mathbf{V}^i$. An interesting question is whether nice theories can be found capturing $\square_i^p, i \geq 2$ using Σ_1^1 definability.

A basic problem is to determine, given a combinatorial principle, what is the least complexity class $\mathbf{C}$ in (37) such that the principle can be proved in the theory associated with $\mathbf{C}$. For example, apparently $\mathbf{C} = \mathbf{TC^0}$ for the pigeonhole principle PHP, and the theory is $\mathbf{VTC}^0$, although it remains open whether PHP can be proved in the system $\mathbf{V^0}(2)$ associated with the smaller class $\mathbf{AC^0}(2)$. In general, it follows that the principle also has polynomial-size proofs in the propositional system corresponding to $\mathbf{C}$ ($\mathbf{TC^0}$-Frege, in the case of PHP). For most natural principles the converse seems to hold, since the propositional proofs are sufficiently uniform that they correspond to one proof in the associated theory. We hope that the theories we present here are sufficiently nice that new positive results will be presented as proofs in these theories, rather than as proofs in the propositional systems (which demonstrate a weaker result).

Besides the PHP, another interesting principle comes from linear algebra: If A and B are $n \times n$ matrices over a field $\mathbb{F}$, then

$$AB = I \supset BA = I \tag{38}$$

If $\mathbb{F}$ is finite or the field of rationals, then the least class in (37) for which the associated theory is known to prove (38) is **P**, even though it seems plausible that **NC** should suffice [45].

The principles of interest need not be just $\forall\Sigma_0^B(\mathcal{L})$ statements, such as the PHP and (38), but can be of higher quantifier complexity. For example, the fact that an $n \times n$ matrix A, say over $GF(2)$, either has an inverse, or has a linear dependence among it rows, can be stated as a Σ_1^B formula $\phi(A)$. Then $\phi(A)$ can be witnessed by functions in $\mathbf{NC}^2$, but the weakest theory among those mentioned here that we know proves $\phi(A)$ is $\mathbf{TV}^0$, the one associated with polynomial time [45].

One of the major (and elusive) goals of propositional proof complexity is to prove separation results for propositional proof systems. Such results would imply separation results for the associated theories. So a worthy goal (and apparently one just as interesting) is to concentrate on the formally easier problem of separating the theories. At present this does not seem feasible in cases in which we have been unable to separate the corresponding complexity classes. Our witnessing theorems imply that the Σ_1^B theorems of two theories can be separated by assuming the complexity classes associated with the theories are distinct. This assumption apparently does not suffice to separate the Σ_0^B theorems, but perhaps some other plausible complexity assumption does suffice.

7 Acknowledgments

I would like to thank Antonina Kolokolova, Jan Krajicek, Phuong Nguyen, Steven Perron, and Alan Skelley for helpful comments on a draft of this paper.

References

[1] M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[2] M. Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the IEEE 29th Annual Symposium on Foundations of Compter Science*, pages 346–355, 1988.

[3] T. Arai. A bounded arithmetic *AID* for Frege systems. *Annals of Pure and Applied Logic*, 103:155–199, 2000.

[4] S. Buss. *Bounded Arithmetic.* Bibliopolis, 1986.

[5] S. Buss. The Boolean formula value problem is in ALOGTIME. *Proceedings of the 19th Annual ACM Symposium on Theory of Computating (STOC'87)*, pages 123–131, 1987.

[6] S. Buss. Polynomial size proofs of the Propositional Pigeonhole Principle. *J. Symbolic Logic*, 52:916–927, 1987.

[7] S. Buss. Axiomatizations and conservation results for fragments of bounded arithmetic. In *Logic and Computation, Proceedings of a Workshop held at Carnegie Mellon University*, pages 57–84. AMS, 1990.

[8] S. Buss. The Graph of Multiplication is Equivalent to Counting. *Information Processing Letters*, 41:199–201, 1992.

[9] S. Buss. First-order proof theory of arithmetic. In S. Buss, editor, *Handbook of Proof Theory*, pages 79–147. Elsevier, 1998. Available on line at `www.math.ucsd.edu/~sbuss/ResearchWeb/HandbookProofTheory/`.

[10] S. Buss. An introduction to proof theory. In S. Buss, editor, *Handbook of Proof Theory*, pages 1–78. Elsevier, 1998. Available on line at `www.math.ucsd.edu/~sbuss/ResearchWeb/HandbookProofTheory/`.

[11] S. Buss. Polynomial-size frege and resolution proofs of st-connectivity and hex tautologies. Submitted, 2003.

[12] P. Clote and G. Takeuti. Bounded arithmetic for NC, ALogTIME, L and NL. *Annals of Pure and Applied Logic*, 56:73–117, 1992.

[13] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Proceedings of the International Congress Logic, Methodology, and Philosophy of Science*, pages 24–30. North Holland, 1965.

[14] S. Cook. Feasibly constructive proofs and the propositional calculus. *Proceedings of the 7th Annual ACM Symposium on Theory of computing*, pages 83–97, 1975.

[15] S. Cook. Relating the provable collapse of P to NC^1 and the power of logical theories. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 39:73–91, 1998.

[16] S. Cook. Slides for Edinburgh talk. Available on line at `www.cs.toronto.edu/~sacook`, 2001.

[17] S. Cook. Csc 2429 course notes: Proof complexity and bounded arithmetic. Available on line at `www.cs.toronto.edu/~sacook/csc2429h/`, 2002.

[18] S. Cook and A. Kolokolova. A second-order system for polytime reasoning based on Grädel's theorem. *Annals of Pure and Applied Logic*, 124:193–231, 2003.

[19] S. Cook and A. Kolokolova. A second-order theory for **NL**. submitted, 2004.

[20] S. Cook and T. Morioka. Quantified propositional calculus and a second-order theory for NC^1. submitted, 2004.

[21] S. Cook and N. Thapen. The strength of replacement in weak arithmetic. submitted, 2003.

[22] S. Cook and A. Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63:103–200, 1993.

[23] M. Dowd. Propositional representation of arithmetic proofs. PhD Thesis, University of Toronto, 1979.

[24] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits and the polynomial-time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[25] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Springer, 1993.

[26] A. Ignjatovic and P. Nguyen. Characterizing Polynomial Time Computable Functions Using Theories with Weak Set Existence Principles. In *Computing: The Australasian Theory Symposium*, 2003.

[27] N. Immerman. Nondeterministic space is closed under complementation. In *SCT: Annual Conference on Structure in Complexity Theory*, 1988.

[28] N. Immerman. *Descriptive Complexity*. Springer, 1999.

[29] J. Johannsen and C. Pollett. On the Δ_1^b-bit-comprehension rule. In S. Buss, P. Hájek, and P. Pudlák, editors, *Logic Colloquium 98*, ASL Lecture Notes in Logic, pages 262–279. 2000.

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

[31] J. Krajíček. *Bounded Arithmetic, Propositional Logic and Computational Complexity*. Cambridge University Press, 1995.

[32] J. Krajíček and P. Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift f. Mathematkal Logik u. Grundlagen d. Mathematik*, 36:29–46, 1990.

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

[34] J. Krajíček and G. Takeuti. On bounded Σ_1^1-polynomial induction. In S. Buss and P. Scott, editors, *Feasible Mathematics*. Birkhäuser, 1990.

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

[36] P. Nguyen. *VTC^0: A Second-Order Theory for TC^0*. MSc Thesis, Department of Computer Science, University of Toronto, 2004.

[37] P. Nguyen and S. Cook. VTC^0: A second-order theory for TC^0. manuscript, 2004.

[38] R. Parikh. Existence and feasibility in arithmetic. *Journal of Symbolic Logic*, 36:494–508, 1971.

[39] J. Paris and A. Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic*, number 1130 in Lecture Notes in Mathematics, pages 317–340. Springer, 1985.

[40] P. Pudlák and S. Buss. How to lie without being (easily) convicted and the lengths of proofs in propositional calculus. In *Computer Science Logic '94*, volume 933 of *LNCS*, pages 151–162. Springer-Verlag, 1995.

[41] A. A. Razborov. An equivalence between second order bounded domain bounded arithmetic and first order bounded arithmetic. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 247–77. Oxford University Press, 1993.

[42] A. A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 344–386. Birkhauser, 1995.

[43] A. Skelley. A third-order bounded arithmetic theory for pspace. submitted, 2003.

[44] R. Smullyan. *Theory of Formal Systems*. Princeton University Press, 1961.

[45] M. Soltys and S. Cook. The proof complexity of linear algebra. To appear in Annals of Pure and Applied Logic, 2003.

[46] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26:279–284, 1988.

[47] G. Takeuti. RSUV isomorphism. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–86. Oxford University Press, 1993.

[48] D. Zambella. Notes on polynomially bounded arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.

[49] D. Zambella. End extensions of models of linearly bounded arithmetic. *Annals of Pure and Applied Logic*, 88:263–277, 1997.