Archive for Mathematical Logic manuscript No.
(will be inserted by the editor)

Stephen Cook · Tsuyoshi Morioka

Quantified Propositional Calculus and a Second-Order Theory for $\mathbf{NC}^1$

Received: date / Revised version: date –

Abstract. Let H be a proof system for quantified propositional calculus (QPC). We define the Σ_j^q-witnessing problem for H to be: given a prenex Σ_j^q-formula A, an H-proof of A, and a truth assignment to the free variables in A, find a witness for the outermost existential quantifiers in A. We point out that the Σ_1^q-witnessing problems for the systems G_1^* and G_1 are complete for polynomial time and **PLS** (polynomial local search), respectively.

We introduce and study the systems G_0^* and G_0, in which cuts are restricted to quantifier-free formulas, and prove that the Σ_1^q-witnessing problem for each is complete for $\mathbf{NC}^1$. Our proof involves proving a polynomial time version of Gentzen's midsequent theorem for G_0^* and proving that G_0-proofs are $\mathbf{TC}^0$-recognizable. We also introduce QPC systems for $\mathbf{TC}^0$ and prove witnessing theorems for them.

We introduce a finitely axiomatizable second-order system $\mathbf{VNC}^1$ of bounded arithmetic which we prove isomorphic to Arai's first order theory $\mathbf{AID}$ + Σ_0^b-CA for uniform $\mathbf{NC}^1$. We describe simple translations of $\mathbf{VNC}^1$ proofs of all bounded theorems to polynomial size families of G_0^* proofs. From this and the above theorem we get alternative proofs of the $\mathbf{NC}^1$ witnessing theorems for $\mathbf{VNC}^1$ and **AID**.

1. Introduction

Quantified Propositional Calculus (QPC) is obtained by introducing quantifiers into propositional calculus, where $(\exists x)A(x)$ is equivalent to $A(\mathtt{T}) \vee (A(\mathtt{F})$ and $(\forall x)A(x)$ is equivalent to $A(\mathtt{T}) \wedge A(\mathtt{F})$. Krajíček and Pudlák [27,26] introduced the Gentzen-style sequent calculus proof system G for QPC, together with the hierarchy of fragments

$$G_1^* \subseteq G_1 \subseteq G_2^* \subseteq G_2 \subseteq \ldots$$

Here G_i is G restricted so that only Σ_i^q or Π_i^q formulas can occur in proofs, where a formula is in Σ_i^q if it has a prenex form with at most $i-1$ alternations of quantifiers, beginning with $\exists$, and dually for Π_i^q. G_i^* is G_i restricted to treelike proofs.

Stephen Cook: Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, Ontario, M5S3G4 CANADA

Tsuyoshi Morioka: Department of Computer Science, University of Toronto, 10 King's College Road, Toronto, Ontario, M5S3G4 CANADA

Key words: Bounded Arithmetic – Computational Complexity – Proof Complexity

The systems are related to the polynomial hierarchy (**PH**) in that the decision problem for validity of Σ_i^q sentences is complete for the level $\mathbf{\Sigma_i^P}$ of **PH**, and similarly for Π_i^q and $\mathbf{\Pi_i^P}$.

The systems G_i and G_i^* are also closely related to Buss's hierarchy of theories

$$S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq ...$$

of bounded arithmetic. In particular, G_i simulates proofs of Σ_i^b formulas in T_2^i [27] and G_i^* simulates proofs of Σ_i^b formulas in S_2^i [26]. Subsequently Pollett [31] introduced the systems L_i^* based on G_i^* which are related to the bounded arithmetic theories R_2^i in the same way.

We modify the definitions of G_i and G_i^* by allowing arbitrary QPC formulas in proofs, but restricting cut formulas to be Σ_i^q and restricting the target formulas in $\exists$-right and $\forall$-left rules to be quantifier-free. We prove that the modified systems are polynomially equivalent to the original for proving $\Sigma_i^q \cup \Pi_i^q$ formulas. Their advantages are that they are complete systems for proving all valid QPC formulas, and the quantifier introduction rules always increase quantifier complexity. As a result of the modification, we obtain G_0 and G_0^* as new and interesting QPC systems, which are polynomially equivalent to Frege systems when proving quantifier-free theorems.

For $j \geq 1$, we define the Σ_j^q-witnessing problem for a QPC system H to be: given a prenex Σ_j^q-formula A, an H-proof of A, and a truth assignment to the free variables of A, find a witness for the outermost existential quantifiers in A. We point out that results of Krajíček and Pudlák on the provability of the reflection principles for the QPC systems [27] and witnessing theorems for bounded arithmetic [26] show that, for $i \geq 1$, the Σ_i^q-witnessing problems for G_i^* and G_i are complete for $\mathbf{FP}^{\Sigma_{i-1}^p}$ and $\mathbf{PLS}^{\Sigma_{i-1}^p}$, respectively, where $\mathbf{FP}$ is the class of polytime functions and $\mathbf{PLS}$ (Polynomial Local Search) is essentially the class of optimization problems solvable by local search algorithms [23].

Our main interest is in the systems G_0 and G_0^*, and their relationship to the the second-order theory $\mathbf{VNC}^1$ introduced here. In particular, we show that the Σ_1^q-witnessing problems for both G_0^* and G_0 are complete for the class $\mathbf{FNC}^1$ of $\mathbf{NC}^1$-functions. (In this paper we use $\mathbf{NC}^1$ to mean **Dlogtime-uniform** $\mathbf{NC}^1$, which is the same as the class **Alogtime** of problems accepted in time $O(\log n)$ on an alternating Turing machine; see [2].) Our proof uses Buss's $\mathbf{NC}^1$ algorithm for the Boolean Formula Value Problem [4, 7], and involves showing that G-proofs are recognizable in uniform $\mathbf{TC}^0$.

We show how to extract a propositional "Herbrand disjunction" from a G_0 proof π of a prenex formula, and show that this disjunction has Frege proofs of size polynomial in $|\pi|$. This result is used in the above witnessing theorem, and also in proving a polynomial time version of Gentzen's midsequent theorem for G_0^*. We use these techniques to prove that G_0^* p-simulates G_0 for Σ_1^q formulas, something that does not hold for G_1^* versus G_1, unless **PLS** is contained in **FP**.

We consider propositional systems which allow threshold gates, and extend them to systems with propositional quantifiers. In particular we define for $d = 1, 2, ...$ the system $TG_0(d)$, which allows cuts only on quantifier-free formulas, and in which all quantifier-free formulas in a proof have depth $\leq d$. We prove that the $T\Sigma_1^q$-witnessing problem for $TG_0(d)$ can be solved by a **TC**^{0}-function, and conversely every **TC**^{0}-function is reducible to such a witnessing problem for some d.

We introduce a second-order system **VNC**^{1} of bounded arithmetic which is inspired by Arai's [1] first-order theory **AID** for **Alogtime**. Our second-order treatment results in a substantial simplification of both the description of the theory and the proofs of the main theorems. **VNC**^{1} is obtained by extending the theory **V**^{0} (essentially $I\Sigma_0^{1,b}$) of $\mathbf{AC^0}$ reasoning by adding a scheme Σ_0^B-$TreeRec$ for tree recursion, based on the heap data structure. We prove that the Σ_1^B-definable functions in **VNC**^{1} are precisely those in **FNC**^{1}, and then prove that **VNC**^{1} is RSUV isomorphic to **AID** + Σ_0^b-CA, where Σ_0^b-CA is the comprehension scheme for sharply-bounded formulas. As a corollary, we obtain that Arai's theory Σ_0^b-RD is equivalent to **AID** + Σ_0^b-CA, a result not mentioned in [1].

We go on to describe translations of **VNC**^{1} proofs of all bounded theorems (not just Σ_0^B theorems) to polynomial size families of G_0^* proofs. This translation generalizes and is much simpler than the translation of Σ_0^b-theorems of **AID** to polynomial size Frege proofs given in [1]. From this and the above main theorem (witnessing for G_0^*) we get alternative proofs of the **FNC**^{1} witnessing theorems for **VNC**^{1} and **AID**.

1.1. Organization

This paper is organized as follows. Section 2 introduces QPC systems G_i and G_i^* and other basic definitions. In Section 3 we define the notions of π-prototypes and the Herbrand π-disjunction for a G_0-proof π, and we prove a polynomial-time version of Gentzen's midsequent theorem for G_0^*. In Section 4 we describe our propositional witnessing problem and prove that the Σ_1^q-witnessing problems for both G_0 and G_0^* are complete for the class of **NC**^{1}-functions. In Section 5 we extend a sequent calculus for propositional threshold logic into quantified threshold calculi for **TC**^{0} and show that the witnessing problems for these systems characterize **TC**^{0}. Section 6 is an exposition on the syntax and semantics of second-order theories and complexity classes. In Section 7 we describe the theory **VNC**^{1} and its relationship to **NC**^{1} and the first-order theory **AID**. Section 8 contains the propositional translations from **VNC**^{1} to G_0^*. Section 9 concludes with remarks on relevant issues and open problems.

2. Quantified Propositional Calculus

Let $\mathtt{T}$ and $\mathtt{F}$ denote the truth values $true$ and $false$, respectively. Quantified Propositional Calculus (QPC) is obtained by introducing quantifiers into propositional calculus, where $(\exists x)A(x)$ is equivalent to $A(\mathtt{T}) \vee A(\mathtt{F})$ and $(\forall x)A(x)$ is equivalent to $A(\mathtt{T}) \wedge A(\mathtt{F})$.

Let $\{p_i : i \in \mathbb{N}\}$ and $\{x_i : i \in \mathbb{N}\}$ be the sets of p-variables and x-variables, respectively. We use the p-variables to denote free variables and the x-variables to denote bound variables: see Definition 2 below.

Definition 1. Formulas *and their* outer connectives *are defined inductively as follows. (1) The atomic formulas are* ($\mathtt{T}$)*,* ($\mathtt{F}$)*, and* (p_i) *and* (x_i) *for every* $i \in \mathbb{N}$*. (2) If* ϕ *and* ψ *are formulas, then so are* $(\phi \wedge \psi)$*,* $(\phi \vee \psi)$*, and* $(\neg\phi)$*. The outer connective of these formulas are* $\wedge$*,* $\vee$*, and* $\neg$*, respectively. (3) If* ϕ *is a formula, then for every* $i \in \mathbb{N}$*, both* $(\exists x_i \phi)$ *and* $(\forall x_i \phi)$ *are formulas. The outer connective of these formulas are* $\exists x_i$ *and* $\forall x_i$*, respectively.*

Often we do not write all the parentheses. Note that we parenthesize the atomic formulas since it somewhat simplifies parsing operations for QPC in Section 4.3.

Definition 2. *A formula* A *is said to be* proper *and called a* QPC formula *iff every occurrence of an* x*-variable in* A *is bound. A formula that is proper and quantifier-free is called* propositional*.*

The validity of QPC formulas is defined in an obvious way.

Both Σ_0^q and Π_0^q denote the set of propositional formulas. For $i \geq 1$, Σ_i^q is the set of QPC formulas that has a prenex form with at most $i-1$ quantifier alternations beginning with $\exists$, and Π_i^q is the dual of Σ_i^q. Note that $\Sigma_{i-1}^q \subseteq \Pi_i^q$ and $\Pi_{i-1}^q \subseteq \Sigma_i^q$ for all $i \geq 1$.

The following definitions are from [27], which generalize those of [20] for propositional proof complexity. Let V be some set of valid QPC formulas. A polytime computable function Q that maps $\{0,1\}^*$ onto V is called a *quantified proof system for* V, and we say that π is a Q-proof of $Q(\pi)$. The following is an easy generalization of a fundamental theorem of propositional proof complexity by Cook and Reckhow [20], connecting the question of proof lengths to open problems of complexity theory:

Theorem 1. *(i) There exists a proof system* Q *in which every valid QPC formula* A *has a proof of size polynomial in* $|A|$ *iff* $\mathbf{NP} = \mathbf{PSPACE}$*. (ii) For every* $i \geq 0$*, there exists a proof system* Q *in which every valid* Σ_i^q*-formula* A *has a proof of size polynomial in* $|A|$ *iff* $\mathbf{NP} = \mathbf{\Pi^P_{i+1}}$*.*

Note that $\mathbf{NP} = \mathbf{\Pi^P_{i+1}}$ for some $i \geq 0$ if and only if $\mathbf{NP} = \mathbf{PH}$.

Let Q_1 and Q_2 be quantified proof systems. We say that Q_2 *p-simulates* Q_1 iff there exists a polytime function f that, if π_1 is a Q_1-proof of A, then $f(\pi_1)$ is a Q_2-proof of A. We say that Q_1 and Q_2 are p-equivalent if they p-simulate each other.

Let PK denote the Gentzen-style sequent calculus for propositional logic of [9,16]. The initial sequents of PK are $\mathtt{F} \rightarrow$ and $\rightarrow \mathtt{T}$ and $A \rightarrow A$ for any propositional formula A. It has structural rules (weakening, contraction, exchange), the cut rule which derives $\Gamma \rightarrow \Delta$ from two sequents $A, \Gamma \rightarrow \Delta$ and $\Gamma \rightarrow \Delta, A$, and propositional rules that introduce new connectives into the sequents. For example, the $\wedge$-left rule derives $A \wedge B, \Gamma \rightarrow \Delta$ from $A, B, \Gamma \rightarrow \Delta$, and the $\wedge$-right rule takes two upper sequents $\Gamma \rightarrow \Delta, A$ and $\Gamma \rightarrow \Delta, B$ and derives $\Gamma \rightarrow \Delta, A \wedge B$. The rules that take two upper sequents, i.e., cut, $\vee$-left, and $\wedge$-right, are called *binary inference rules*; all the other rules are called *unary inference rules*. For each inference rule, the *principal formulas* are the formulas in the lower sequent to which the rule is applied. For example, the principal formula of $\wedge$-right above is

$A \wedge B$. The exchange rules are the only rules with two principal formulas, and cut has no principal formula. The *auxiliary formulas* of a rule are the formulas in the upper sequents to which the rule is applied. For example, the auxiliary formulas of $\wedge$-right are A and B. The weakening rules do not have any auxiliary formula.

A PK-proof is a sequence $S_1, \ldots, S_k$ such that each sequent S_i is either an initial sequent or is derived from at most two preceding sequents. S_k is called the *endsequent* and a formula in S_k is referred to as an *endformula*. Tree-like PK is PK with the restriction that, in a proof, each sequent occurs as an upper sequent of an inference step at most once. We write $|\pi|$ to denote the *size* of proof π, which is the total number of symbols in π. We say that a family of sequents has polynomial-size PK-proofs if there exists a polynomial p such that, for every formula A in the family, there exists a PK-proof of A whose size is at most $p(|A|)$.

In [27], Krajíček and Pudlák introduced Gentzen-style sequent calculus systems for QPC which we call KPG, KPG_i, and KPG_i^*, for $i \geq 0$. KPG is obtained by augmenting PK with the following new inference rules:

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

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

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. For each of the above quantifier rules, the auxiliary formula is the formula that occurs only in the upper sequent (i.e., either $A(B)$ or $A(b)$) and the principal formula is the formula that appears only in the lower sequent (i.e., either $\forall x A(x)$ or $\exists x A(x)$). Finally, the sequent $A \rightarrow A$ for any QPC formula A is allowed as an initial sequent.

For $i \geq 1$, KPG_i is obtained by requiring that all formulas in a KPG_i-proof be $\Sigma_i^q \cup \Pi_i^q$. KPG_i^* is KPG_i restricted to treelike proofs.

The restriction that KPG_i and KPG_i^* can only reason about Σ_i^q-formulas seems artificial. Moreover, the known correspondences between T_2^i and KPG_i and between S_2^i and KPG_i^* (Theorem 2 below) are not optimal; T_2^i can reason about formulas with more than $i-1$ quantifier alternations while T_2^i-proofs of such formulas cannot, by definition, be translated into polysize KPG_i-proofs.

We remedy this shortcoming by modifying the definition of KPG_i and KPG_i^* to obtain the systems which we call G_i and G_i^*.

Definition 3. *G is obtained by augmenting PK with the four quantifier-introduction rules, with the additional restriction that the target of every $\forall$-left and $\exists$-right step be quantifier-free.*

Definition 4. *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.*

The restriction that the target of $\forall$-left and $\exists$-right rules of G be quantifier-free means that all quantifier-introduction rules increase the quantifier complexity of the auxiliary formula, as opposed to those rules of KPG which can result in a decrease of the quantifier complexity. The definition of G_i and G_i^* by restricting the complexity of cut formulas is in the spirit of traditional proof theory, and it is motivated by the way Pitassi defines the bounded-depth propositional PK system by restricting the depth of cut formulas [30].

Lemma 1 below shows that our systems are natural extensions of Krajíček and Pudlák's systems. The advantages of our systems are that they are complete proof systems for the whole QPC.

Lemma 1. *G and KPG are p-equivalent. Moreover, for every $i \geq 1$, KPG_i and KPG_i^* are p-equivalent to G_i and G_i^*, respectively, for proving valid $\Sigma_i^q \cup \Pi_i^q$-formulas.*

Proof. We prove that KPG_i and G_i are p-equivalent for every $i \geq 0$. The proof is identical for KPG_i^* versus G_i^* and KPG versus G.

KPG_i obviously p-simulates G_i with respect to proving valid $\Sigma_i^q \cup \Pi_i^q$-formulas. For the other direction, it suffices to show that G_i can simulate $\exists$-right and $\forall$-left steps in KPG_i with quantified targets. For the $\exists$-right case, let S be the sequent $\Gamma \rightarrow \Delta, \exists x A(x)$ which is derived from $\Gamma \rightarrow \Delta, A(B)$ with B quantified. G_i can derive S from $A(B) \rightarrow \exists x A(x)$ and $\Gamma \rightarrow \Delta, A(B)$ by weakening and cut on $A(B)$. G_i can cut $A(B)$ since, by the definition of KPG_i, $A(B)$ is $\Sigma_i^q \cup \Pi_i^q$. It remains to show that $A(B) \rightarrow \exists x A(x)$ has short proofs in G_i. This sequent is derived from $B, A(B) \rightarrow \exists x A(x)$ and $A(B) \rightarrow \exists x A(x), B$ by cut on B, and these sequents in turn follow from (T1) and (T2) of Lemma 2 below by $\exists$-right with atomic targets.

The simulation of $\forall$-left inference of KPG_i by G_i follows in an analogous way from (T3) and (T4) of Lemma 2 below. □

Lemma 2. *Let $\exists x A(x)$ and B be QPC formulas and $A(B)$ be the result of substituting B for all occurrences of x in $A(x)$. The following four sequents have cut-free G_0^*-proofs of size $O(|A(B)|^2)$:*
(T1): $B, A(B) \rightarrow A(\top)$,
(T2): $A(B) \rightarrow A(\bot), B$,
(T3): $B, A(\top) \rightarrow A(B)$, and
(T4): $A(\bot) \rightarrow B, A(B)$.

Proof. Simultaneous induction on the structure of $A(x)$. □

In fact, the proof of Lemma 1 shows that there is a polytime function that converts KPG_i-proofs into G_i-proofs in which *the target of every $\exists$-right and $\forall$-left step is atomic*, and it follows that the restriction of atomic targets does not decrease the power of G_i for proving valid $\Sigma_i^q \cup \Pi_i^q$-formulas. However, we allow nonatomic targets in Definitions 3 and 4 because we do not know whether we can restrict the targets to be atomic without weakening the power of G_i for proving formulas with more than i quantifier alternations. Note that the proof of Lemma 1 requires cuts on the auxiliary formulas of $\exists$-right and $\forall$-left, and if the endsequent contains a formula not in $\Sigma_i^q \cup \Pi_i^q$, then G_i may not be able to cut these auxiliary formulas.

Let π be a G-proof. Free variables of π that occur in the endsequent are called *parameter variables*. Below we follow Buss [3,9] in defining the *free variable normal form* of proofs, which is slightly stronger than Takeuti's *regular proofs* [35].

Definition 5. *([3, 9]) Let π be a tree-like G-proof. We say that π is in* free variable normal form *if the following conditions are met: (i) no parameter variable is used as an eigenvariable; and (ii) every nonparameter variable is used as an eigenvariable exactly once in π.*

If π is a tree-like proof in free variable normal form, it follows that, for every nonparameter variable b, the sequents containing an occurrence of b form a subtree of π whose root is the upper sequent of the inference in which b is used as the eigenvariable. Any tree-like proof can be converted into free variable normal form by renaming bound variables and replacing nonparameter variables b with the logical constant $\mathtt{T}$ if b is never used as an eigenvariable. Throughout this paper, we assume that all tree-like QPC proofs are in free variable normal form.

Krajíček and Pudlák in [27] define a translation of a bounded formula $A(a)$ of first-order bounded arithmetic into a polynomial size family $\{||A(a)||[n]\}_{n\in\mathbb{N}}$ of QPC formulas and prove the following:

Theorem 2. *([27, 26]) For $i \geq 1$, if A is a Σ_i^b theorem of T_2^i then the corresponding family $\{||A(a)||[n]\}_{n\in\mathbb{N}}$ of valid Σ_i^q-formulas has polynomial-size G_i-proofs which can be constructed in time polynomial in n. Similarly for S_2^i and G_i^*.*

The above result is tight with respect to the quantifier complexity of A, since its proof does not work for bounded theorems of T_2^i or S_2^i that are not Σ_i^q.

Nonetheless, any bounded theorem of T_2^i (or S_2^i) can be translated into the corresponding valid QPC formulas with polysize proofs using the second-order translations described in Section 8.

Theorem 3. *Theorem 2 continues to hold when A is a Σ_j^b theorem of T_2^i (respectively S_2^i) for any $j \geq 0$.*

Proof. (Sketch) Theorem 23 states this result for the second-order isomorphic images $\mathbf{TV}^i$ and $\mathbf{V}^i$ of T_2^i and S_2^i, respectively. □

3. The Polynomial-time Midsequent Theorem for G_0^* and G_0

Note that KPG_0 and KPG_0^* are quantifier-free propositional proof systems (PK and tree-like PK, respectively), and thus our G_0 and G_0^* are new quantified proof systems for the whole QPC that have never been studied. Since the cut formulas of G_0 and G_0^* are quantifier-free, they are similar to first-order theories T axiomatized by purely universal formulas, all of whose theorems have LK derivations with all cuts on quantifier-free formulas [9], and therefore it is reasonable to attempt to obtain for G_0 and G_0^* the counterparts of the proof-theoretic statements regarding such theories T. Of course, a major difference between the proof theory for T and the study of G_0 and G_0^* is that the former is concerned with the existence of proofs and various normal forms for proofs without much interest in the size of proofs or the complexity of converting proofs into normal forms, while for G_0 and G_0^* the size and complexity are the major concern.

Gentzen's midsequent theorem for first-order LK states that, if S is a valid sequent consisting of prenex formulas only, then S has a tree-like cut-free LK-derivation with a 'midsequent' S' such that all quantifier inferences occur below S' and all propositional inferences take place above S'. Krajíček pointed out in [26] that the midsequent theorem holds for G. We prove below a polynomial-time version of the midsequent theorem for G_0^* which is useful in the study of proof complexity of G_0^*.

Definition 6. *Suppose that π is a G_0-proof with endsequent $\rightarrow A$, where A is a quantified QPC formula in prenex form. Then any quantifier-free formula A' in π that occurs as the auxiliary formula of a quantifier-introduction step is called a* π-prototype *of A. We define the* Herbrand π-disjunction *to be the sequent*

$$\rightarrow A_1, \ldots, A_m$$

where $A_1, \ldots, A_m$ are all the π-prototypes of A.

Assume that π is a G_0-proof of the sequent $\rightarrow A$ where A is of the form

$$Q_1 x_1 \ldots Q_k x_k F(\mathbf{p}, x_1, \ldots, x_k) \tag{1}$$

with $Q_i \in \{\exists, \forall\}$ for each $i \in [1, k]$ and F quantifier-free. Then, for every π-prototype A' of A, there exists a unique sequence $B_1, \ldots, B_k$ of propositional formulas such that

$$A' =_{syn} F(\mathbf{p}, B_1, \ldots, B_k).$$

Intuitively, each B_i is either the target of the $\exists$-right step or the eigenvariable of the $\forall$-right step that introduces the bound variable x_i into A'.

Definition 7. *Let π, A, A', and $B_1, \ldots, B_k$ be as in the preceding paragraph. For each $i \in [1, k]$, we call B_i* the ith component of A'.

The following is similar to a general form of Herbrand's theorem for first-order logic by Buss [9]. However, since our claim is simpler, so is the proof.

Lemma 3. *Let π be a G_0-proof of a sequent $\rightarrow A$ with A a quantified QPC formula in prenex form. Then the Herbrand π-disjunction is valid, and it has a PK-proof of size polynomial in $|\pi|$.*

Proof. Assume that π is the sequence $S_1, \ldots, S_k$ of sequents, where S_k is $\rightarrow A$. For every $i \in [1, k]$, if sequent S_i is $\Gamma_i \rightarrow \Delta_i$, then define S_i' to be $\Gamma_i \rightarrow \Delta_i'$, where Δ_i' is obtained from Δ_i by removing all quantified formulas and adding all π-prototypes $A_1, \ldots, A_m$. Note that Γ_i contains no quantified formula, and S_k' is the Herbrand π-disjunction.

We argue that every S_i' has a PK-proof of size polynomial in $|\pi|$ by induction on i. If S_i contains no quantified formula, then $S_i =_{syn} S_i'$ and there is nothing to prove. Assume that S_i contains a quantified formula. The only nontrivial case is when S_i is derived from S_j which does not contain a quantified formula, and this happens only in weakening or quantifier introduction. In either case, S'_i follows from S'_j by introducing $A_1, \ldots, A_m$ by weakening. □

We emphasize that the truth of Lemma 3 depends crucially on the fact that all cut formulas in G_0-proofs are quantifier-free. Our proof does not work if cuts on quantified formulas are present in π.

Below we state and prove a polynomial-time version of the midsequent theorem for G_0^*. That every G_0^*-proof of a prenex formula can be transformed into a midsequent proof is standard. The novel part of our statement is to explicitly point out that the transformation can be done in polynomial-time.

Theorem 4. *(The Polynomial-time Midsequent Theorem for G_0^*) Let π be a G_0^*-proof of sequent S of the form $\to A$ with quantified prenex formula A. Then there exists another G_0^*-proof π' of S such that:*
(i) π' starts with a quantifier-free derivation of the Herbrand π-disjunction S_π; and
(ii) only contraction, exchange, $\forall$-right, and $\exists$-right inference steps occur between S_π and the endsequent.

In fact, there is a polynomial-time algorithm that converts π into such π'.

Proof. Let A be of the form

$$Q_1x_1 \ldots Q_kx_kF(\mathbf{p}, x_1, \ldots, x_k)$$

with F quantifier-free and $Q_i \in \{\exists, \forall\}$ for each $i \in [1, k]$. Let $A_1, \ldots, A_m$ be all the π-prototypes. By Lemma 3, the Herbrand π-disjunction

$$S_\pi =_{syn} \quad \to A_1, \ldots, A_m$$

has a short PK-proof. It suffices to derive $\to A$ from S_π using contraction, exchange, $\forall$-right and $\exists$-right rules only.

By the properties of π-prototypes, for each $j \in [1, m]$, π-prototype A_j is of the form

$$A_j =_{syn} F(\mathbf{p}, B_1^j, \ldots, B_k^j),$$

where $B_1^j, \ldots, B_k^j$ are the components of A_j. For each B_i^j, we define its *elimination step* in π to be the quantifier-introduction step that introduces Q_ix_i into a descendant of A_j. It is possible for one elimination step to be associated with c components $B_{i_1}^{j_1}, \ldots, B_{i_c}^{j_c}$. This happens iff $i_1 = i_2 = \cdots = i_c$ and the descendants of $A_{j_1}, \ldots, A_{j_c}$ are contracted in π at some point before the elimination inference occurs.

Let s and t be inference steps in π. We say that an inference step *s precedes t in π* if s occurs in the subproof of π ending with t. We first try to derive $\to A$ from S_π by introducing quantifiers into the π-prototypes consistently with the way the quantifier-introduction steps occur in π so that each π-prototype is transformed into a copy of A. More specifically, we execute a quantifier-introduction step only after executing all quantifier-introduction steps that precede it in π. This procedure works except for the cases in which there is a $\forall$-right inference that is associated with two components b_i^j and b_i^v, both of which are an eigenvariable b. Since b occurs in two distinct formulas in the current sequent, $\forall$-right rule is applicable neither to the formula containing b_i^j nor the one containing b_i^v. Because π is in free variable normal form, both b_i^j and b_i^v have the same eliminating inference, and we know that this happens only when the descendants of A_j and A_v are contracted in π before the elimination inference occurs. Thus, if we not only introduce the quantifiers but also contract quantified formulas consistently with the way these steps occur in π, we will be able to derive $\rightarrow A$ from S_π. We incorporate this into the above procedure as follows. Define a Q-inference to be a quantifier-introduction step or a contraction step of two quantified formulas. Starting with S_π, we execute each Q-inference s of π (interleaved with exchange steps as needed) only after all the Q-inferences that precede s are executed.

Obviously such a sequence of Q-inferences exists because π exists. It suffices to show that the eigenvariable condition for $\forall$-right is never violated. Assume, for the sake of contradiction, that the eigenvariable condition is violated when we apply $\forall$-right rule on $b_i^j =_{syn} b$. This violation is because b occurs as some b_v^u in another formula of the sequent. It follows that the elimination inference of b_v^u does not precede that of b_i^j in π, and hence the $\forall$-right step on b_v^u occurs outside the subproof rooted at the $\forall$-right on b_i^j. But this violates the conditions of free variable normal form. □

In Section 4.3 we show that many parsing operations for G-proofs are in $\mathbf{TC}^0$. Based on this fact, we can lower the complexity of constructing π' from polynomial-time to $\mathbf{TC}^0$; more details will be in [28].

After obtaining the above proof of Theorem 4 we realized that Gentzen's argument for his original LK midsequent theorem can also be used to prove our version via elimination of cuts on quantifier-free B by introducing $B \vee \neg B$ in the antecedent. However, as remarked above, our construction of π' can be carried out in $\mathbf{TC}^0$ while it is not obvious whether the Gentzen-style construction can be done below $\mathbf{P}$.

Our proof above does not work if π is not tree-like, since in such π there can be two $\forall$-right steps s and t on two formulas B and B' with the same eigenvariable b such that neither precedes the other in π. Once B and B' are in the same sequent, which will be the case if we try to derive $\rightarrow A$ from the Herbrand π-disjunction, the $\forall$-right inferences on b is applicable to neither B nor B'. However, if all quantifier-introduction steps in π are $\exists$-right, then this problem never arises, and thus we can prove the following weaker statement for G_0:

Theorem 5. *Theorem 4 holds for G_0 if the endformula A is prenex Σ_1^q.*

Proof. Since the endformula is Σ_1^q, all quantifier-introduction steps in π are $\exists$-right. From the Herbrand π-disjunction $\rightarrow A_1, \ldots, A_m$ we can easily derive a sequent containing m copies of A by repeated applications of $\exists$-right rule. □

The above construction of a midsequent proof can also be shown to be in $\mathbf{TC}^0$.

It is shown by Krajíček that tree-like PK p-simulates dag-like PK [26], and therefore G_0^* p-simulates G_0 for propositional tautologies. Based on this and the above results, we obtain a stronger p-simulation for G_0^* and G_0.

Theorem 6. *G_0^* p-simulates G_0 for proving prenex Σ_1^q-formulas.*

Proof. Let π be a G_0-proof of a sequent S containing one prenex Σ_1^q-formula. Apply Theorem 5 to obtain π' that proves S from the Herbrand π-disjunction S_π. Since the subproof π_1 of π' rooted at S_π is a PK-proof, and tree-like PK p-simulates PK [26], we can convert π_1 into a tree-like PK-proof π_2 with only polynomial size increase. Finally, replace π_1 in π' with π_2; the result is a G_0^*-proof of S. $\square$

Theorem 6 is interesting because we later show that a similar p-simulation for G_1 and G_1^* implies $\mathbf{PLS} = \mathbf{FP}$ (see Theorem 7), which is not believed to be true. It is open whether Theorem 6 can be generalized to a p-simulation of G_0 by G_0^* for proving all prenex formulas. Since we are not able to prove Theorem 4 for G_0, the above argument does not work for this general case.

4. The Witnessing Problems for QPC

Let $i \geq 0$ and let H be either G_i or G_i^*. For $j \geq 1$, define the Σ_j^q-witnessing problem for H, written $Witness[H, \Sigma_j^q]$, as follows. The input is $(\pi, \mathbf{v})$, where π is an H-proof of a Σ_j^q-formula $A(\mathbf{p})$ of the form

$$A(\mathbf{p}) =_{syn} \exists x_1 \ldots \exists x_k F(\mathbf{p}, x_1, \ldots, x_k) \tag{2}$$

with F prenex Π_{j-1}^q, and $\mathbf{v}$ is a truth assignment to the free variables $\mathbf{p}$. A solution for the problem is a witness for $A(\mathbf{v})$, i.e., a sequence $\mathbf{u}$ of $\mathtt{T}, \mathtt{F}$ such that $F(\mathbf{v}, \mathbf{u})$ holds.

The complexity of Σ_i^q-witnessing problems for G_i and G_i^* follow from the existing results in bounded arithmetic.

Theorem 7. *$Witness[G_1^*, \Sigma_1^q]$ and $Witness[G_1, \Sigma_1^q]$ are complete for $\mathbf{FP}$ and $\mathbf{PLS}$, respectively. More generally, for $i \geq 1$, $Witness[G_i^*, \Sigma_i^q]$ and $Witness[G_i, \Sigma_i^q]$ are complete for $\mathbf{FP}^{\Sigma_{i-1}^p}$ and $\mathbf{PLS}^{\Sigma_{i-1}^p}$, respectively.*

Proof. We first prove that $Witness[G_i^*, \Sigma_i^q]$ is in $\mathbf{FP}^{\Sigma_{i-1}^q}$. Krajicek [26] shows that S_2^i proves the *i-reflection principle* for G_i^*:

$$\forall \mathbf{v} \forall \pi \forall A(\mathbf{p})[(A(\mathbf{p}) \in \Sigma_i^q \wedge \pi : G_i^* \vdash A(\mathbf{p})) \supset A(\mathbf{v}) \text{ is a true } \Sigma_i^q\text{-sentence}] \tag{3}$$

where $\mathbf{v}, \pi, A(\mathbf{p})$ are actually numbers that encode the corresponding truth assignment, proof, and formula, respectively. '$A(\mathbf{v})$ is a true Σ_i^q-sentence' is expressed by a Σ_i^b-formula stating that there exists a witness to the outermost existential quantifiers of $A(\mathbf{v})$. Thus the i-reflection principle is a $\forall\Sigma_i^b$-sentence. The $Witness[G_i^*, \Sigma_i^q]$ is exactly the problem of witnessing (3), which is in $\mathbf{FP}^{\Sigma_{i-1}^p}$ by Buss's witnessing theorem [3] (also in [26,8]).

That $Witness[G_i, \Sigma_i^q] \in \mathbf{PLS}^{\Sigma_{i-1}^p}$ is proven analogously from the facts that T_2^i proves i-reflection principle for G_i [27,26] and that witnessing Σ_i^b-theorems of T_2^i is in $\mathbf{PLS}^{\Sigma_{i-1}^b}$ [12] (also see [13]).

It is shown by Buss in [3] that every $f \in \mathbf{FP}^{\Sigma^p_{i-1}}$ is Σ^b_i-definable in S^i_2; that is, there exists $\phi \in \Sigma^b_i$ such that $\mathbb{N} \models \forall x \phi(x, f(x))$ and $S^i_2 \vdash \forall x \exists y \leq t(x) \phi(x, y)$. Let $A(a)$ denote $\exists y \leq t(a)\phi(a, y)$ with one free variable a. We describe a reduction of f to $Witness[G^*_i, \Sigma^q_i]$. Given $a \in \mathbb{N}$ with $|a| = n$, the Σ^q_i-formula $||A(a)||[n]$ corresponding to A and a G^*_i-proof π of it can be constructed in time polynomial in n by Theorem 2. Let A^q denote $||A(a)||[n]$. Since A^q is not prenex even when A is prenex, we do the following. Let $A^+ \in \Sigma^q_i$ be a prenex form of A^q such that

$$A^q \to A^+$$

has short, cut-free G^*_0-proofs. By a cut on A^q we construct from π another G^*_i-proof π' with endformula A^+. Finally, by letting $\mathbf{v}$ the truth assignment encoding the bits of a, a solution for $Witness[G^*_i, \Sigma^q_i]$ on the instance $(\pi', \mathbf{v})$ gives the value of $f(a)$. Thus, f is many-one reducible to $Witness[G^*_i, \Sigma^q_i]$.

The hardness of $Witness[G_i, \Sigma^q_i]$ for $\mathbf{PLS}^{\Sigma^p_{i-1}}$ is proven in a way completely analogous to the G^*_i case above. □

It follows from Theorem 7 that, if G^*_1 p-simulates G_1, then $\mathbf{FP} = \mathbf{PLS}$, which is believed to be false. More generally, a p-equivalence between any two quantified sequent calculi discussed in this paper implies a collapse of the corresponding complexity classes.

The Σ^b_j-definable search problems (i.e., multifunctions) in T^i_2 and S^i_2 for all $j > i$ have been known to coincide with natural complexity classes [5,25,32], and we conjecture that $Witness[G_i, \Sigma^q_j]$ and $Witness[G^*_i, \Sigma^q_j]$ are complete for these complexity classes. This conjecture is also related to the question whether T^i_2 and S^i_2 prove the j-reflection principles for G_i and G^*_i for $j > i$, respectively.

Cook [16] and Skelley [34] have direct proofs of $Witness[G^*_i, \Sigma^q_i] \in \mathbf{FP}^{\Sigma^p_{i-1}}$ and $Witness[G_i, \Sigma^q_i] \in \mathbf{PLS}^{\Sigma^p_{i-1}}$, respectively, which do not go though the provability of the reflection principles in bounded arithmetic.

4.1. The Σ^q_1-Witnessing Problem for G_0: Witness Formulas

Since G_0 and G^*_0 are new proof systems for QPC, the complexity of their witnessing problems have never been studied. In this and the next few subsections we study the complexity of $Witness[G_0, \Sigma^q_1]$.

Let π be the input G_0-proof of $A(\mathbf{p})$ of the form (2) with F quantifier-free, and let $A_1, \ldots, A_m$ be a sequence of all π-prototypes. Note that the sequent $\to A_1, \ldots, A_m$ is valid by Lemma 3. For each $j \in [1, m]$, define a formula E_j which states that A_j is the first in the sequence $A_1, \ldots, A_m$ that is satisfied, i.e.,

$$E_j =_{syn} (\neg A_1 \wedge \neg A_2 \wedge \ldots \wedge \neg A_{j-1}) \wedge A_j.$$

Thus, any truth assignment $\mathbf{v}$ satisfies E_l for exactly one value $l \in [1, m]$.

Definition 8. *Let $A(\mathbf{p})$ and k be as in (2) and $A_1, \ldots, A_m$ and $E_1, \ldots, E_m$ be as in the above paragraph. For all $i \in [1, k]$, define*

$$\phi_i =_{syn} \bigvee_{j=1}^{m} (E_j \wedge B^j_i),$$

*where B_i^j is the ith component of A_j. We call ϕ_i the i*th π-witness formula.

We prove that the π-witness formulas compute a solution for the Σ_1^q-witnessing problem for G_0 and that this fact has short PK proofs.

Theorem 8. *Let π be a G_0-proof of a prenex Σ_1^q-formula A of the form $\exists x_1 \ldots \exists x_k F(\mathbf{p}, x_1, \ldots, x_k)$, where F is quantifier-free. Let $\phi_1, \ldots, \phi_k$ be the π-witness formulas. Then $F(\mathbf{p}, \phi_1, \ldots, \phi_k)$ is a tautology and it has a PK-proof of size polynomial in $|\pi|$.*

Proof. We write F to denote $F(\mathbf{p}, x_1, \ldots, x_k)$. $F[\mathbf{x}/\boldsymbol{\phi}]$ denotes the result of substituting ϕ_i for x_i, $i \in [1, k]$. Our goal is to show that $F[\mathbf{x}/\boldsymbol{\phi}]$ has PK-proofs of size polynomial in $|\pi|$.

Let $A_1, \ldots, A_m$ be the π-prototypes of A. For each $j \in [1, m]$, define sequents S_j and T_j as

$$\begin{array}{l} S_j =_{syn} \quad \rightarrow F[\mathbf{x}/\boldsymbol{\phi}], A_1, \ldots, A_j \\ T_j =_{syn} A_j \rightarrow F[\mathbf{x}/\boldsymbol{\phi}], A_1, \ldots, A_{j-1} \end{array}$$

S_m is derived by weakening the Herbrand disjunction $\rightarrow A_1, \ldots, A_m$ which, by Lemma 3, has a PK-proof of size polynomial in $|\pi|$. For each $j \in [1, m-1]$, S_j is derived from S_{j+1} and T_{j+1} by cut, and finally $F[\mathbf{x}/\boldsymbol{\phi}]$ is derived from S_1 and T_1 by cut. Thus it suffices to prove that, for each $j \in [1, m]$, the sequent T_j has a polysize PK-proof.

Fix $j \in [1, m]$. Let C be a subformula of F, and let $C[\mathbf{x}/\mathbf{B^j}]$ be the result of substituting, for every $i \in [1, k]$, B_i^j for x_i in C. Note that $C[\mathbf{x}/\mathbf{B^j}]$ is a subformula of A_j. Define two sequents U_1^C and U_2^C as follows:

$$\begin{array}{ll} U_1^C: & C[\mathbf{x}/\mathbf{B^j}], A_j \rightarrow C[\mathbf{x}/\boldsymbol{\phi}], A_1, \ldots, A_{j-1} \\ U_2^C: & C[\mathbf{x}/\boldsymbol{\phi}], A_j \rightarrow C[\mathbf{x}/\mathbf{B^j}], A_1, \ldots, A_{j-1} \end{array}$$

It is clear that T_j follows from U_1^C for $C =_{syn} F$ by contraction. We prove that, for any subformula C of F, both U_1^C and U_2^C have PK-proofs of size polynomial in $|\pi|$.

We proceed by structural induction on C. If C does not contain any occurrence of an x-variable, then $C[\mathbf{x}/\boldsymbol{\phi}]$ and $C[\mathbf{x}/\mathbf{B^j}]$ are identical, and therefore both U_1^C and U_2^C follow from initial sequents by weakening.

For the other base case, suppose that C is an atom (x_i) for $i \in [1, k]$. Then $C[\mathbf{x}/\boldsymbol{\phi}]$ is ϕ_i and $C[\mathbf{x}/\mathbf{B^j}]$ is B_i^j, and hence we need to show that the following two sequents have polysize PK-proofs:

$$U_1^B: \quad B_i^j, A_j \rightarrow \bigvee_{l=1}^{m} (E_l \wedge B_i^l), A_1, \ldots, A_{j-1} \tag{4}$$

$$U_2^B: \quad \bigvee_{l=1}^{m} (E_l \wedge B_i^l), A_j, \rightarrow B_i^j, A_1, \ldots, A_{j-1} \tag{5}$$

The sequent (4) is derived by weakening and $\vee$-right from

$$B_i^j, A_j \rightarrow (E_j \wedge B_i^j), A_1, \ldots, A_{j-1}$$

which follows from the two sequents $B_i^j \to B_i^j$ and $A_j \to E_j, A_1, \ldots, A_{j-1}$. By the definition of E_j, the latter sequent has short proofs.

The sequent (5) is derived by m applications of $\vee$-left from the sequents

$$(E_l \wedge B_i^l), A_j, \to B_i^j, A_1, \ldots, A_{j-1} \tag{6}$$

for each $l \in [1, m]$. We claim that all of the sequent of the form (6) have short PK proofs. If $l < j$, then the sequent contains A_l in both sides of '$\to$'. If $l = j$, then B_i^j appears in both sides. Finally, if $l > j$, then the antecedent of the sequent contains both A_j and $\neg A_j$. This concludes the case where $B =_{syn} (x_i)$ for some $i \in [1, k]$.

The inductive step is straightforward. If C is $(C_1 \vee C_2)$, $(C_1 \wedge C_2)$, or $(\neg C_1)$, then the sequents U_1^C and U_2^C have short PK-proofs from $U_1^{C_1}$, $U_2^{C_1}$, $U_1^{C_2}$, and $U_2^{C_2}$, all of which have short PK-proofs. $\square$

Theorem 8 can be strengthened to say that there is a $\mathbf{TC}^0$ algorithm which, given π, constructs a polynomial-size PK-proof of $F(\mathbf{p}, \phi_1, \ldots, \phi_k)$ [28].

4.2. The Σ_1^q-Witnessing Problem for G_0: An $\mathbf{NC}^1$*-algorithm*

$\mathbf{NC}^1$ is the class of languages accepted by families of bounded fan-in boolean circuits of logarithmic depth and polynomial size. Throughout this paper we write $\mathbf{NC}^1$ to mean **Dlogtime**-uniform $\mathbf{NC}^1$, which is equivalent to the class **Alogtime** of languages accepted by an alternating Turing machine in $O(\log n)$ time. See [2, 22] for more information on $\mathbf{NC}^1$. Let $\Sigma = \{0, 1\}$. A function $f : \Sigma^* \to \Sigma^*$ is an $\mathbf{NC}^1$-function iff $|f(x)| \in |x|^{O(1)}$ and its bit graph

$$R_f(x, i, c) \equiv \text{the } i\text{th bit of } f(x) \text{ is } c$$

is in $\mathbf{NC}^1$, where i is presented in unary. $\mathbf{NC}^1$-functions over an arbitrary finite alphabet Σ is defined similarly by encoding each symbol of Σ as k-bit strings, where k depends only on $|\Sigma|$. $\mathbf{FNC}^1$ is the class of $\mathbf{NC}^1$-functions (also see Section 6.2).

$\mathbf{AC^0}$ is the class of predicates decidable by families of constant-depth, polynomial-size circuits with AND, OR, and NOT gates of arbitrary fan-in, and $\mathbf{TC}^0$ is obtained by allowing *threshold gates*, which output 1 iff the number of 1's in its inputs is at least some threshold [22]. Again we only work with **Dlogtime**-uniform versions of these classes [2]. A function f is $\mathbf{AC^0}$ iff $|f(x)| \in |x|^{O(1)}$ and its bit graph is in $\mathbf{AC^0}$.

In this subsection we prove that there exists an $\mathbf{NC}^1$-function that outputs a solution for $Witness[G_0, \Sigma_1^q]$. By Theorem 8, it suffice to show that, given an instance $(\pi, \mathbf{v})$ of $Witness[G_0, \Sigma_1^q]$, π-witness formulas ϕ_i for any i can be evaluated in $\mathbf{NC}^1$. Since Buss has shown in [4] that the Boolean Formula Value Problem of evaluating the input propositional formula under the given truth assignment is in $\mathbf{NC}^1$, we only need to show that the parsing operations necessary for recognizing occurrences of ϕ_i's in π are in $\mathbf{NC}^1$.

Fix the alphabet Σ_{QPC} such that the inputs to $Witness[G_0, \Sigma_1^q]$ are represented as finite strings over it:

$$\Sigma_{QPC} = \{\mathtt{T}, \mathtt{F}, p, x, 0, 1, (,), \wedge, \vee, \neg, \exists, \forall, \to, comma, \#\},$$

where $comma$ denotes the comma. The symbols 0 and 1 are used to denote the indices of variables. A variable p_i is written as $p^\frown i$ with $i \in \{0,1\}^+$, where $^\frown$ is string concatenation, and similarly for x_i. Formulas and sequents are encoded as strings over Σ_{QPC} without the sharp symbol (#). A G_0-proof is representable as $S_1\#S_2\#\ldots\#S_m$, that is, a sequence of sequents separated by the sharp (#) symbol such that every S_i is either an initial sequent or derived from at most two preceding sequents by an appropriate inference rule. We also fix an encoding scheme for truth assignments.

From now on, Z will always denote a finite string over Σ_{QPC} and $n = |Z|$. For $1 \le i \le n$, α_i is the ith symbol of Z, i.e.,

$$Z =_{syn} \alpha_1\alpha_2\ldots\alpha_n.$$

For $1 \le i \le j \le n$, we write $Z[i,j]$ to denote the substring $\alpha_i\ldots\alpha_j$ of Z. If $j < i$ then $Z[i,j]$ is empty.

Theorem 9. *The Σ_1^q-witnessing problem for G_0 is solvable by an* **NC**^{1}*-function.*

Proof. It suffices to describe an $\mathbf{Alogtime}$ Turing machine M for computing the π-witness formulas. The input to M is a tuple (Z, i, c), and M accepts iff the ith witness formula evaluates to $c \in \{\mathtt{T}, \mathtt{F}\}$. In Lemmas 4, 5, and 8 below, we prove that formulas, G_0-proofs, and π-prototypes are **TC**^{0}-recognizable. Fix an ordering on the π-prototypes to be the order in which they appear in Z. Then recognizing the kth component of the lth π-prototype is also in **TC**^{0} by Lemma 8. Note that, since **TC**^{0} $\subseteq$ $\mathbf{Alogtime}$, M can evaluate any **TC**^{0} predicates.

Given a string Z as input, M accepts iff there exists $r \in [1,n]$ such that all of the following hold: (i) $Z[1,r]$ encodes a G_0-proof π of A of the form (2) with F quantifier-free; (ii) $Z[r+1,n]$ encodes a truth assignment $\mathbf{v}$ to the parameter variables of π; and (iii) the ith π-witness formula ϕ_i evaluates to c under $\mathbf{v}$. The first two conditions involve **TC**^{0} predicates only.

It remains to describe how M can evaluate ϕ_i. M existentially guesses j and verifies that $j \ge m$. Since π-prototypes are **TC**^{0}-recognizable, computing m is a **TC**^{0}-function and therefore M can compute m. Then M universally verifies that B_i^j, A_j, and $\neg A_l$ for every $l < j$ are true under $\mathbf{v}$. The Boolean Formula Value Problem is in $\mathbf{Alogtime}$ [4,7], and therefore M can evaluate all these formulas. Finally, note that in every computation path M only needs to guess indices of Z, each of which has only $\lceil \log n \rceil$ bits. Thus M runs in alternating time $O(\log n)$. $\square$

The next subsection is devoted to proving that the predicates for parsing that are used in the above proof are indeed **TC**^{0}.

4.3. Parsing Operations for QPC are in $\mathbf{TC}^0$

Let $R \subseteq \Sigma_{QPC}^* \times (\mathbb{N})^k$ for $k \geq 0$. R is in $\mathbf{AC^0}$ iff R is representable by a first-order (FO) formula [2,21], where FO is defined as follows. The terms of FO are constructed from constants for the natural numbers, n (which denotes the length of the string input) and variables using addition and multiplication. The predicate symbols of FO are $=$, $<$, and $Syn_\sigma(i)$ for each $\sigma \in \Sigma_{QPC}$. Variables denote indices of the input string of length n, and therefore they range over $\{0, \ldots, n\}$. First-order logic with majority quantifier (FOM) is obtained by allowing a special quantifier M such that $Mx\phi(x)$ means that $\phi(x)$ is true for more than half of the possible x's. R is in $\mathbf{TC}^0$ iff it is representable by an FOM-formula [2,21]. The following is an easy and useful fact: if $\phi(a)$ is an FOM-formula with one free variable a, then there exists another FOM-formula $\psi(b)$ such that, for every k, the sentence $\psi(k)$ holds iff there are k values of x such that $\phi(x)$ holds.

Buss in [6] shows that $\mathbf{NC}^1$ can parse propositional formulas and also recognize Frege proofs. His proofs actually show that these can be done in $\mathbf{TC}^0$. Using the same idea, we can prove the following Lemma.

Lemma 4. *The following predicates are in* $\mathbf{TC}^0$*:*
(1) $Formula(Z, i, j) \equiv Z[i, j]$ *is a formula;*
(2) $QPCF(Z, i, j) \equiv Z[i, j]$ *is a QPC formula; and*
(3) $Sequent(Z, i, j) \equiv Z[i, j]$ *is a sequent.*

Proof. (Sketch) For (1), it suffices to ensure that (1A) $Z[i, j]$ is correctly parenthesized and that (1B) there is no substring of $Z[i, j]$ of length 2 that is impossible in a formula. Examples of impossible substrings are px, $x)$, $\neg\exists$, etc. Condition (1A) holds iff (a) $Z[i, j]$ contains an equal number of occurrences of '(' and ')'; and (b) for all $u \in [i, j-1]$, $Z[i, u]$ contains more '(' than ')'. It is easy to see that both (1A) and (1B) are FOM-expressible.

For (2), we only need to check that $Z[i, j]$ is a formula and that for every occurrence of an x-variable x_k in $Z[i, j]$ there is a subformula of $Z[i, j]$ containing the occurrence of x_k and whose outer connective is either $\exists x_k$ or $\forall x_k$. This is FOM-expressible.

We omit a proof for (3), which is easily seen to be FOM-expressible. □

Define $Inf_1(S_i, S_j)$ to be true iff sequent S_i is derivable from sequent S_j by a unary inference rule, where S_i, S_j are given as strings over Σ_{QPC}. Similarly, $Inf_2(S_i, S_j, S_k)$ is true iff S_i is derivable from two sequents S_j and S_k by a binary inference rule.

Lemma 5. *Both* Inf_1 *and* Inf_2 *are* FOM*-expressible.*

Proof. For Inf_2, we need to express that S_i follows from the hypotheses S_j and S_k by cut, $\wedge$-right, or $\vee$-left. The case for $\wedge$-right is handled by verifying that (i) the three sequents involved are identical except for the rightmost formulas; and (ii) the principal formula of S_i is identical to the disjunction of the auxiliary formulas of S_j and S_k, and clearly this is expressible in FOM. The case for $\vee$-left is handled in an analogous manner, and the case for cut is also easily expressible in FOM.

For Inf_1, it is easy to see that there is an FOM formula expressing that S_i is derived from S_j by structural rules or unary propositional rules. Finally, By Lemma 7 below, the property 'S_i follows from S_j by a quantifier rule' is FOM-expressible. $\square$

In order to prove Lemma 7 on which the truth of Lemma 5 depends, we introduce the notion of *identifier* of a subformula. Let A be a formula. For each subformula B of A, we define its identifier (ID) as the string over $\{1, 2\}$ that uniquely determines its location within A as a path from the root of A to the root of B, thinking of A as a tree.

Definition 9. *Let A be a formula. For every subformula B of A, $ID_A(B)$ is defined inductively as follows. (1) $ID_A(A)$ is the empty string ϵ; (2) if B is $(B_1 \odot B_2)$ with $\odot \in \{\wedge, \vee\}$, then $ID_A(B_1)$ is $ID_A(B)^\frown 1$ and $ID_A(B_2)$ is $ID_A(B)^\frown 2$; and (3) if B is either $(\neg B_1)$ or (QxB_1), then $ID(B_1)$ is $ID_A(B)^\frown 1$.*

For example, if A is $(\exists x_1(B \vee (\neg C)))$, then $ID_A(B) = 11$ and $ID_A(C) = 121$.

Lemma 6. *The following is a* **TC**^{0}*-function: given Z, i, j, a, b such that $i \leq a < b \leq j$ and $Z[i, j]$ and $Z[a, b]$ encode formulas A and B, respectively, output $ID_A(B)$.*

Proof. It suffices to show that the bit graph of $ID_A(B)$ is in **TC**^{0}. First, the ith symbol of $ID_A(B)$ is nonempty iff there exists l, m satisfying (i) $Z[l, m]$ is a subformula of A; (ii) B is a subformula of $Z[l, m]$; and (iii) the number of '(' in $Z[i, l-1]$ minus the number of ')' in $Z[i, l-1]$ is equal to i.

Suppose that the ith bit of $ID_A(B)$ is nonempty. Then the ith bit is '2' iff $\alpha_{l-1} \in \{\wedge, \vee\}$. $\square$

Lemma 7. *Define the following predicates to be true iff S_i can be derived from S_j by the corresponding quantifier rule:*
(1) $ExistsLeft(S_i, S_j)$;
(2) $ExistsRight(S_i, S_j)$;
(3) $ForAllLeft(S_i, S_j)$; and
(4) $ForAllRight(S_i, S_j)$.
These predicates are FOM-expressible:

Proof. We informally describe an FOM formula ϕ expressing $ExistsRight$. ϕ is the conjunction $\phi_1 \wedge \phi_2 \wedge \phi_3$ of three FOM formulas. ϕ_1 expresses that S_i and S_j are identical except for their rightmost formulas, and ϕ_2 is true iff the outer connective of the rightmost formula of S_i is $\exists x_k$ for some k. Let P and A be the rightmost formulas of S_i and S_j, respectively (P and A stand for 'principal' and 'auxiliary'). ϕ_3 expresses the following: there exist a propositional subformula B of A and an x-variable x_k such that $A =_{syn} C[x_k/B]$ and $P =_{syn} (\exists x_k C)$. This is true iff, for every subformula ψ' of C, there exists a subformula ψ of A with $ID_C(\psi') = ID_A(\psi)$ such that:

- if ψ' is the atomic formula (x_k), then ψ is B;

- if ψ' is atomic but not (x_k), then ψ and ψ' are identical; and
- if ψ' is not atomic, then ψ and ψ' have the same outer connective.

Note that the above conditions are expressible in FOM.

An FOM formula for $ForAllLeft$ is constructed similarly. For $ExistsLeft$ and $ForAllRight$, we construct FOM formulas in an analogous manner with an additional condition that the subformula B of A be the atomic formula (p_k) such that the free variable p_k does not appear in S_i, which is easily expressible in FOM. □

Finally we are able to show that G_0-proofs are recognizable in $\mathbf{TC}^0$. Define the predicate $Proof_{G_0}(Z,i,j)$ to be true iff the following three conditions hold: (A) $Z[i,j]$ is of the form $S_1\#\ldots\#S_m$, where each S_i is a sequent; (B) for each sequent S_k, either S_k is an initial sequent or it is derived from at most two sequents that precede S_k; and (C) for each sequent S_k there exists $S_{k'}$ with $k<k'$ that can be derived using S_k as an upper sequent. The condition (C) is not strictly necessary, but we add it to ensure that the proof does not contain sequents that are not used in proving the endsequent.

Lemma 8. *The following predicates are FOM-expressible:*
(1) $Proof_{G_0}(Z,i,j)$;
(2) $Prototype(Z,i,j,u,v)$, which holds iff $Z[i,j]$ is a G_0-proof π of a prenex QPC formula and $Z[u,v]$ is a π-prototype; and
(3) $Component(Z,i,j,k,l)$, which is true iff $Z[i,j]$ is the kth component of the lth π-prototype A_l.

Proof. (Sketch) That $Proof_{G_0}$ is FOM-expressible readily follows from the fact that Inf_1 and Inf_2 are FOM-expressible (Lemma 5). Both $Prototype$ and $Component$ are expressible using $Proog_{G_0}$ and $ExistsRight$ and $ForAllRight$. □

We have proven that G_0-proofs are $\mathbf{TC}^0$-recognizable. In fact, this is easily extended to the recognizability of any QPC sequent calculus proofs:

Theorem 10. *Let H be any of G, G_i, or G_i^* for some $i\geq 0$. Then H-proofs are $\mathbf{TC}^0$-recognizable.*

Proof. The only difference between G-proofs and G_0-proofs is that cuts on any QPC formulas are allowed in G, and thus modifying Inf_2 yields the $\mathbf{TC}^0$-recognizability of G-proofs. Similarly, for any $i\geq 1$, G_i-proofs are $\mathbf{TC}^0$-recognizable since $\Sigma_i^q\cup\Pi_i^q$-formulas are $\mathbf{TC}^0$-recognizable.

The tree-like proofs are $\mathbf{TC}^0$-recognizable if we modify the encoding scheme in the following way. If $S_1,\ldots,S_k$ is a G_i^*-proof, then it is encoded as $(S_1,w_1)\#(S_2,w_2)\#\ldots\#(S_k,w_k)$, where each w_i indicates the upper sequents that are used to derive S_i if S_i is not an initial sequent. □

Our proof easily generalizes to the $\mathbf{TC}^0$-recognizability of first-order sequent calculus proofs whose underlying language and nonlogical axioms are $\mathbf{TC}^0$-recognizable. This fact has been known to Buss and possibly a few others [10], but, as far as we know, it has not been explicitly stated in print.

TC^0 is widely believed to be the smallest complexity class in which counting is possible. Since parsing operations requires counting in general, apparently **TC**^{0} is the smallest class containing those parsing operations.

4.4. *Hardness of the Σ_1^q-Witnessing Problem for G_0^**

Let F and G be two functions. We say that F is many-one $\mathbf{AC^0}$-reducible to G if there exist two $\mathbf{AC^0}$-function g, h such that $F(x) = g(G(h(x)))$. It is easy to generalize this definition to the case in which one or both of F and G are total multifunctions.

A function F is said to be *hard for* **FNC**^{1} *under many-one* $\mathbf{AC^0}$*-reductions* iff every **NC**^{1}-function is many-one $\mathbf{AC^0}$-reducible to it. F is *complete for* **FNC**^{1} if F itself is in **FNC**^{1}.

The hardness of $Witness[G_0^*, \Sigma_1^q]$ for **NC**^{1} under many-one $\mathbf{AC^0}$-reductions can be proven as follows. For every propositional sentence A, the sequent

$$\rightarrow (\exists x)[(x \wedge A) \vee (\neg x \wedge \neg A)]$$

has a G_0^*-proof π consisting of k sequents, where k is some constant that does not depend on A. Moreover, it can be shown that π is constructible from A by an $\mathbf{AC^0}$-function. Finally, note that the witness to x is the truth value of A and that the problem of evaluating propositional sentences is hard for **NC**^{1} by Buss's result [4,7].

The fact that $Witness[G_0^*, \Sigma_1^q]$ is hard for **NC**^{1} does not immediately imply its hardness for **FNC**^{1}; however, we can use essentially the same idea to prove the **FNC**^{1} hardness.

Theorem 11. $Witness[G_0^*, \Sigma_1^q]$ *is hard for* **FNC**^{1} *under many-one* $\mathbf{AC^0}$*-reductions.*

Proof. Let f be an arbitrary **NC**^{1}-function, and assume without loss of generality that there is a polynomial p such that $f : \{0,1\}^n \rightarrow \{0,1\}^{p(n)}$ for every n. Since the bit graph $R_f(x, i, c)$ is in **NC**^{1}, and since every **NC**^{1}-predicate is computed by a **Dlogtime**-uniform family of polynomial-size propositional formulas [2], there exists a **Dlogtime**-uniform family $\{A_n\}_n$ of polynomial-size propositional formulas such that, for each $n = |x|$, $A_n(x, i)$ is true if the ith bit of $f(x)$ is 1 and $A_n(x, i)$ is false otherwise, where x and i are represented in A_n by sequences $\mathbf{p}$ and $\mathbf{q}$ of propositional variables, respectively.

Let $m = p(n)$ and, for each $i \in [1, m]$, let $\mathbf{i}$ denote the truth assignment to $\mathbf{q}$ representing i in unary. Define sequent S_n as follows:

$$\rightarrow (\exists y_1) \dots (\exists y_m)[(y_1 \leftrightarrow A_n(\mathbf{p}, \mathbf{1})) \wedge \dots \wedge (y_m \leftrightarrow A_n(\mathbf{p}, \mathbf{m}))]$$

where '$y_i \leftrightarrow A_n(\mathbf{p}, \mathbf{i})$' abbreviates $(y_i \wedge A_n(\mathbf{p}, \mathbf{i})) \vee (\neg y_i \wedge \neg A_n(\mathbf{p}, \mathbf{i}))$. Suppose that π_n is a G_0^*-proof of S_n and that $\mathbf{v}$ is a truth assignment to $\mathbf{p}$ encoding $x \in \{0,1\}^n$. It is easy to see that there is an $\mathbf{AC^0}$ function that computes $f(x)$ given the solution for $Witness[G_0^*, \Sigma_1^q]$ on $(\pi_n, \mathbf{v})$. Thus, it suffices to show the existence of an $\mathbf{AC^0}$-function g such that $g(x)$ is a G_0^*-proof $\pi_{|x|}$ of $S_{|x|}$ of size polynomial in $|x|$.

Below we give an informal description of the proof π_n. The sequent S_n is derived by m applications of $\exists$-right from

$$\to (A_n(\mathbf{p},\mathbf{1}) \leftrightarrow A_n(\mathbf{p},\mathbf{1})) \wedge \ldots \wedge (A_n(\mathbf{p},\mathbf{m}) \leftrightarrow A_n(\mathbf{p},\mathbf{m}))$$

which follows by applications of $\wedge$-right from the sequents

$$\to (A_n(\mathbf{p},\mathbf{i}) \leftrightarrow A_n(\mathbf{p},\mathbf{i})) \tag{7}$$

for each $i \in [1,m]$. It is easy to see that every sequent of the form (7) has a G_0^*-proof with a constant number of sequents, and this completes the description of π_n. Finally, an $\mathbf{AC^0}$-function can output π_n because each line of π_n has a highly uniform structure and it is easy to determine what the jth sequent of π_n should look like for any j. $\square$

From Theorems 9 and 11 we conclude the following:

Theorem 12. *Both* $Witness[G_0, \Sigma_1^q]$ *and* $Witness[G_0^*, \Sigma_1^q]$ *are complete for* $\mathbf{FNC}^1$ *under many-one* $\mathbf{AC^0}$*-reduction.*

5. Quantified Propositional Calculi for $\mathbf{TC^0}$

In this section we sketch sequent calculus systems for $\mathbf{TC}^0$. By taking advantage of the fact that many parsing operations are in $\mathbf{TC}^0$, we obtain a witnessing theorem for these $\mathbf{TC}^0$ sequent calculi similar to Theorem 9 for G_0.

We describe below the sequent calculus system PTK for propositional threshold logic by Buss and Clote [11] with minor modifications. The connectives of PTK are the negation $\neg$ and the unbounded fan-in threshold connectives Th_k for $k \geq 0$. Here $Th_k(A_1, \ldots, A_n)$ holds iff the number of true inputs is at least k. Note that $Th_k(A_1, \ldots, A_n)$ for $k = 1$ and $k = n$ are the $\bigvee_{i=1}^n A_i$ and $\bigwedge_{i=1}^n A_i$, respectively.

The initial sequents of PTK are:
(i) $\to \mathtt{T}$ and $\mathtt{F} \to$ and $A \to A$ for any formula A;
(ii) $Th_k() \to$ for $k \geq 1$; and
(iii) $\to Th_0(A_1, \ldots, A_n)$ for $n \geq 1$.

The structural rules of PTK are: weakening, contraction, exchange, and permutation of the arguments of Th in a formula. PTK has cut, $\neg$-left, $\neg$-right, and the following introduction rules for Th_k with $k \geq 1$:

$$Th_k\text{-left:}\frac{Th_k(A_2, \ldots, A_n), \Gamma \to \Delta \qquad A_1, Th_{k-1}(A_2, \ldots, A_n), \Gamma \to \Delta}{Th_k(A_1, \ldots, A_n), \Gamma \to \Delta}$$

$$Th_k\text{-right:}\frac{\Gamma \to \Delta, A_1, Th_k(A_2, \ldots, A_n) \qquad \Gamma \to \Delta, Th_{k-1}(A_2, \ldots, A_n)}{\Gamma \to \Delta, Th_k(A_1, \ldots, A_n)}$$

Let A be a formula of PTK. The *depth* of A is the maximum number of nestings of connectives in A.

Quantified Threshold Calculus (QTC) is obtained by introducing quantifiers in PTK, with the convention that the x-variables are used for bound variables and the p-variables denote free variables. For $i \geq 0$, define $T\Sigma_i^q$ to be the class of Σ_i^q-formulas over the connectives $\neg$ and Th_k for $k \geq 0$.

Definition 10. *TG is obtained by augmenting PTK with the quantifier-introduction rules. We require that the target of a $\exists$-right and a $\forall$-left be quantifier-free. TG_0 is TG with cuts only on quantifier-free formulas. For $d \geq 1$, $TG_0(d)$ is TG_0 with a restriction that all quantifier-free formulas in a proof be of depth $\leq d$.*

The $T\Sigma_1^q$-witnessing problem for $TG_0(d)$, written $Witness[TG_0(d), T\Sigma_1^q]$, is defined similarly to $Witness[G_0, \Sigma_1^q]$.

Theorem 13. *For every $d \geq 1$, $Witness[TG_0(d), T\Sigma_1^q]$ is solved by some* $\mathbf{TC}^0-function$.

Proof. Fix $d \geq 1$. We can define π-prototypes, Herbrand π-disjunction, and the witness formulas for $TG_0(d)$ and $Witness[TG_0(d), T\Sigma_1^q]$ analogously to those for G_0.

Let $True(A, \mathbf{v})$ be the **NC**^{1} predicate which holds iff the propositional formula A evaluates to $\mathtt{T}$ under the truth assignment $\mathbf{v}$. Note that the proof of Theorem 9 actually shows that there is an FOM formula Φ for evaluating the ith π-witness formula provided that $True(A, \mathbf{v})$ is allowed to appear as atomic formulas.

By making the following two modifications to Φ, we can construct an FOM formula Φ' (with no additional **NC**^{1} predicate) evaluating the ith π-witness formula for $Witness[TG_0(d), T\Sigma_1^q]$.

The first modification is that we replace the predicate $True$ in Φ with an FOM formula $True_d^{PTK}$, which holds iff the input formula A is a (quantifier-free) PTK-formula of depth at most d and A evaluates to $\mathtt{T}$ under the given assignment $\mathbf{v}$. Using the methods in [2], we can prove that, for each $d \in \mathbb{N}$, evaluating a PTK-formula of depth $\leq d$ is in **TC**^{0}, and the existence of $True_d^{PTK}$ follows.

Second, the definition of $ID_A(B)$ should be modified as follows. Let $m = \lceil \log |A| \rceil$ and let $ID_A(A) = \epsilon$. If $B =_{syn} \neg C$, then $ID_A(C) = ID_A(B)^\frown\delta$ with $\delta \in \{0,1\}^m$ is 1 in binary. If $B =_{syn} Th_k(C_1, \ldots, C_n)$, then $ID_A(C_j)$ for $j \in [1, n]$ is $ID_A(B)^\frown\gamma$, where $\gamma \in \{0,1\}^m$ represents j in binary. It is clear that ID_A under this definition is still **TC**^{0}-computable. The rest of the construction of Φ' goes through. □

Theorem 14. *For Every* **TC**^{0}*-function F, there exists $d \in \mathbb{N}$ such that F is reducible to $Witness[TG_0(d), T\Sigma_1^q]$ under many-one* $\mathbf{AC^0}$*-reduction.*

Proof. This is proven analogously to Theorem 11, using the fact that every **TC**^{0} predicate is computed by a **Dlogtime**-uniform family $\{T_n\}_n$ of polynomial-size PTK formulas [2]. □

6. Second order theories

6.1. Syntax and semantics

Our "second order" theories are really two-sorted first order predicate calculus theories, and are based on the elegant syntax of Zambella [37]. 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 sets of natural numbers (which represent binary strings).

The language $\mathcal{L}_A^2$ extends the language of Peano Arithmetic, and consists of the function and predicate symbols $0, 1, +, \cdot, |\,|; \in, \leq, =_1, =_2$. Here $0, 1, +, \cdot$ are function symbols for numbers, and are intended to have their usual interpretation on $\mathbb{N}$. The function symbol $|X|$ denotes 1 plus the largest element in X, or 0 if X is empty (roughly the length of the corresponding string). $t \in X$ denotes set membership, but we usually use the notation $X(t)$ for $t \in X$, since we think of $X(t)$ as the t-th bit of the string X. Finally $=_1$ and $=_2$ denote equality on numbers and strings, respectively, but we will drop the subscripts, since they will be clear from context.

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 $t = u$, $X = Y$, $t \leq u$, $t \in X$ 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.

$\Sigma_0^B = \Pi_0^B$ is the set of all formulas in $\mathcal{L}_A^2$ 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 defined dually. 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 $\Pi_i^{1,b}$ defined in standard treatments because we require that all string quantifiers are in front.

6.2. Second order complexity classes

Our basic complexity classes are classes of relations $R(\mathbf{x}, \mathbf{Y})$, where each x_i in the list $\mathbf{x}$ ranges over $\mathbb{N}$ and each Y_i in the list $\mathbf{Y}$ ranges 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, and each finite subset input is presented by the corresponding bit string. Thus **P** is the class of such relations accepted in polynomial time on a Turing machine. By $\mathbf{NC}^1$ we mean uniform $\mathbf{NC}^1$ or **Alogtime** (alternating log time). By $\mathbf{AC^0}$ we mean uniform $\mathbf{AC^0}$, or **LH** (the log time hierarchy). The following result ([21] and [16] pp 54–55) nicely connects $\mathbf{AC^0}$ and our second order language $\mathcal{L}_A^2$.

Lemma 9. *A relation $R(\mathbf{x}, \mathbf{X})$ is in $\mathbf{AC^0}$ iff it is represented by some Σ_0^B-formula $\phi(\mathbf{x}, \mathbf{X})$.*

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

Definition 11. *The* bit graph B_F *of a string function* F *is defined by*

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

If **C** *is a second order 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**.

6.3. The theory $\mathbf{V}^0$

The base theory $\mathbf{V}^0$ [18, 16] (called Σ_0^p-$comp$ in [37]) is associated with the complexity class $\mathbf{AC^0}$, and all second order 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. $x + 1 \neq 0$	**B8.** $(x \leq y \wedge y \leq x) \supset x = y$
B2. $x + 1 = y + 1 \supset x = y$	**B9.** $0 + 1 = 1$
B3. $x + 0 = x$	**B10.** $0 \leq x$
B4. $x + (y + 1) = (x + y) + 1$	**B11.** $x \leq y \wedge y \leq z \supset x \leq z$
B5. $x \cdot 0 = 0$	**B12.** $x \leq y \vee y \leq x$
B6. $x \cdot (y + 1) = (x \cdot y) + x$	**B13.** $x \leq y \leftrightarrow x < y + 1$
B7. $x \leq x + y$	**B14.** $x \neq 0 \supset \exists y \leq x(y + 1 = x)$
L1. $X(y) \supset y < \|X\|$	**L2.** $y + 1 = \|X\| \supset X(y)$

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

The Σ_0^B comprehension scheme is

$$\Sigma_0^B\text{-COMP:} \qquad \exists X \leq y \forall z < y(X(z) \leftrightarrow \phi(z, \mathbf{x}, \mathbf{Y})) \tag{8}$$

where $\phi(z, \mathbf{x}, \mathbf{Y})$ is any Σ_0^B formula not containing X.

A result in [18] shows that $\mathbf{V}^0$ is finitely axiomatizable.

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 a minimization scheme, and the induction formula

$$[X(0) \wedge \forall y < z(X(y) \supset X(y + 1))] \supset X(z) \tag{9}$$

(See [18] or [16] for details.) From this and Σ_0^B-COMP we have

Theorem 15. $\mathbf{V}^0$ *proves the scheme*

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

where $\phi(x)$ *is any* Σ_0^B*-formula (possibly containing free variables other than* x*).*

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, see [8]). That $\mathbf{V}^0$ is an extension of $\mathbf{I\Delta_0}$ is immediate from the 2-$BASIC$ axioms and Theorem 15. 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) [16].

We use as a pairing function the term

$$\langle x, y\rangle =_{def} (x+y)(x+y+1)+2y \tag{10}$$

Then $\mathbf{V}^0$ 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 $X(\mathbf{x})$. It is easy to see that $\mathbf{V}^0$ proves the analog of Σ_0^B-COMP (8) for multidimensional arrays.

If we think of Z as a two-dimensional array, then we can represent row x in this array by $Z^{[x]}$ [37], where $G(x, Z) = Z^{[x]}$ is the $\mathbf{FAC^0}$ string function with bit-defining axiom

$$Z^{[x]}(i) \leftrightarrow i < |Z| \wedge Z(x, i) \tag{11}$$

We can add this string function $Z^{[x]}$ together with its defining equation (11) to form a conservative extension of $\mathbf{V}^0$.

Definition 12. *Let T be a theory extending $\mathbf{V}^0$. A string function $F(\mathbf{x}, \mathbf{X})$ is Σ_1^B-* definable *in T if there is a Σ_1^B-formula ϕ such that*

$$Y = F(\mathbf{x}, \mathbf{X}) \leftrightarrow \phi(\mathbf{x}, \mathbf{X}, Y) \text{ and}$$
$$T \vdash \forall \mathbf{x} \forall \mathbf{X} \exists! Y \phi(\mathbf{x}, \mathbf{X}, Y)$$

The Σ_1^B-definability for a number function $f(\mathbf{x}, \mathbf{X})$ is defined similarly.

Theorem 16. *A function (string or number) is Σ_1^B-definable in $\mathbf{V}^0$ iff it is in* $\mathbf{FAC^0}$.

Proof. That every function in $\mathbf{FAC^0}$ is Σ_1^B-definable in $\mathbf{V}^0$ follows from Lemma 9 and Definition 11 of $\mathbf{FAC^0}$, using the axiom scheme Σ_0^B-COMP (8). The converse follows from Theorem 17 below, where T is the conservative extension of $\mathbf{V}^0$ resulting from introducing for each Σ_0^B-formula $\phi(i, \mathbf{x}, \mathbf{X})$ and term $t(\mathbf{x}, \mathbf{X})$ a function F with defining axiom (12). □

Definition 13. *(Witnessing) Let T be a theory over a language $\mathcal{L}$ which includes $\mathcal{L}_A^2$ and let $\exists \mathbf{Y} \le \mathbf{t}\phi(\mathbf{x}, \mathbf{X}, \mathbf{Y})$ be a $\Sigma_1^B(\mathcal{L})$-formula. Then T* witnesses *the formula if there are function symbols $\mathbf{F}$ in $\mathcal{L}$ such that*

$$T \vdash \mathbf{F}(\mathbf{x}, \mathbf{X}) \le \mathbf{t} \wedge \phi(\mathbf{x}, \mathbf{X}, \mathbf{F}(\mathbf{x}, \mathbf{X}))$$

Theorem 17. *(Witnessing) Suppose T is a theory which extends $\mathbf{V}^0$, and is defined over a language $\mathcal{L}$ and suppose that for every $\Sigma_0^B(\mathcal{L})$-formula $\phi(i, \mathbf{x}, \mathbf{X})$ and term $t(\mathbf{x}, \mathbf{X})$ of $\mathcal{L}_A^2$ there is a function symbol F in $\mathcal{L}$ such that T proves*

$$F(\mathbf{x}, \mathbf{X})(i) \leftrightarrow i < t \wedge \phi(i, \mathbf{x}, \mathbf{X}) \tag{12}$$

Suppose further that each axiom of T is Σ_1^B and witnessed by T. Then every $\Sigma_1^B(\mathcal{L})$ theorem of T is witnessed by T.

Proof. (sketch) By cut elimination, every Σ_1^B theorem of T has a normal form LK proof from the axioms of T (an "anchored" proof [8]) in which every formula is Σ_1^B. By induction on the length of such proofs it follows that the formulas in each line of the proof can be witnessed in T (in a suitable sense). [8, 16]. □

7. The theory VNC^1

We define the system **VNC**^{1} by adding a tree recursion axiom scheme Σ_0^B-$TreeRec$ to $\mathbf{V}^0$. This scheme is intended to take the place of the predicates $A^{\ell,B,\overline{D},I}$ and their defining axioms in Arai's theory **AID** [1], which captures reasoning in **Alogtime** (uniform **NC**^{1}). Our scheme is a simplified second order version of Arai's Σ_0^b-RD ([1] Definition 7.1), using the idea of the heap data structure.

The Σ_0^B-$TreeRec$ scheme is

$$\begin{aligned}&\exists Z \le 2a \forall i < a[(Z(i+a) \leftrightarrow \psi(i)) \wedge \\ &0 < i \supset (Z(i) \leftrightarrow \phi(i)[Z(2i), Z(2i+1)])]\end{aligned} \tag{13}$$

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 to be 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, $0 < i \le 2a-1$. The leaves of the tree are indexed by any i such that $a \le i \le 2a-1$ and leaf number i is assigned value $\psi(i)$. The internal nodes of the tree are indexed by any i such that $1 \le i \le a-1$, 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.

For Σ_0^B formulas $\phi(i, \mathbf{x}, \mathbf{X})[p,q]$ and $\psi(i, \mathbf{x}, \mathbf{X})$ in the Σ_0^B-$TreeRec$ scheme (13) we define the Σ_0^B formula $B^{\phi,\psi}(a, \mathbf{x}, \mathbf{X}, Z)$ to be the part of (13) which comes after $\exists Z \le 2a$. That is,

$$\begin{aligned}B^{\phi,\psi}(a, \mathbf{x}, \mathbf{X}, Z) \;\equiv\; &\forall i < a[(Z(i+a) \leftrightarrow \psi(i)) \wedge \\ &0 < i \supset (Z(i) \leftrightarrow \phi(i)[Z(2i), Z(2i+1)])]\end{aligned} \tag{14}$$

Lemma 10. *For all Σ_0^B formulas ϕ, ψ*

$$\mathbf{VNC}^1 \vdash \exists Z! \le 2a B^{\phi,\psi}(a, \mathbf{x}, \mathbf{X}, Z) \tag{15}$$

Proof. Existence of Z follows from (13). Uniqueness can be proved in $\mathbf{V}^0$ using Σ_0^B-IND. □

7.1. Defining $\mathbf{NC}^1$ *relations and functions in* $\mathbf{VNC}^1$

As usual, we define a uniform $\mathbf{NC}^1$ relation to be one in **Alogtime**. Here $\mathbf{NC}^1$ always refers to uniform $\mathbf{NC}^1$.

We start by showing how to define $\mathbf{NC}^1$ relations in $\mathbf{VNC}^1$. Every formula $B^{\phi,\psi}$ (14) defines a relation $R^{\phi,\psi}$ (computed by the recursion scheme (13)) with defining axiom

$$R^{\phi,\psi}(a,i,\mathbf{x},\mathbf{X}) \leftrightarrow \exists Z \leq 2a(B^{\phi,\psi}(a,\mathbf{x},\mathbf{X},Z) \wedge Z(i)) \tag{16}$$

Lemma 11. *The relation $R^{\phi,\psi}$ is in $\mathbf{NC}^1$, for each pair $\phi[p,q],\psi$ of Σ_0^B formulas.*

Proof. By Lemma 9, each Σ_0^B-formula represents an $\mathbf{AC^0}$ relation, which is therefore in $\mathbf{Alogtime}$. To prove the lemma, it suffices to show there exists an indexed alternating Turing machine M with inputs $(a,i,\mathbf{x},\mathbf{X})$ (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 starts by guessing the binary notation for the input i, and verifying its guess in time $O(\log n)$ using its indexed access to the input tape. It then guesses that $Z(i)$ is true, and verifies its guess by recursively guessing and verifying $Z(j)$ for various values of j. 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 $Z(j) \leftrightarrow \psi(j-a,\mathbf{x},\mathbf{X})$, 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 $Z(j) \leftrightarrow \phi(j,\mathbf{x},\mathbf{X})[Z(2j),Z(2j+1)]$, (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 (13), which is $O(\log a) = O(\log n)$. □

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 (16). 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 12. *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.

Conversely, we appeal to Theorem 3.1 of [1], which states that every $\mathbf{NC}^1$ relation is $\Sigma_0^b(\mathcal{L}_{\mathbf{AID}})$-definable in **AID**. We argue in the proof of Theorem 20 (RSUV isomorphism) that the $\Sigma_0^b(\mathcal{L}_{\mathbf{AID}})$ formulas correspond to the $\Sigma_0^B(\mathcal{L}_{TreeRec})$ formulas. □

We denote by $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$ the theory whose language is $\mathcal{L}_{TreeRec}$ and whose axioms are those of $\mathbf{VNC}^1$ together with the defining axioms (16). Clearly $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$ is a conservative extension of $\mathbf{VNC}^1$. By $\Sigma_0^B(\mathcal{L}_{TreeRec})$-COMP we mean the scheme (8), where ϕ is any $\Sigma_0^B(\mathcal{L}_{TreeRec})$ formula.

Lemma 13. **VNC**$^1(\mathcal{L}_{TreeRec})$ *proves the* $\Sigma_0^B(\mathcal{L}_{TreeRec})$*-COMP scheme.*

Proof. First note that **VNC**^{1} Δ_1^B-defines each relation $R^{\phi,\psi}$, since the Σ_1^B formula representing $R^{\phi,\psi}$ in (16) is provably equivalent to a Π_1^B formula. That is, from (15) it follows that **VNC**^{1} proves

$$\exists Z \le 2a(B^{\phi,\psi}(a,\mathbf{x},\mathbf{X},Z) \wedge Z(i)) \leftrightarrow \forall Z \le 2a((B^{\phi,\psi}(a,\mathbf{x},\mathbf{X},Z) \supset Z(i))$$

The lemma would follow easily from this and the Σ_0^B-COMP axioms if **VNC**^{1} proves the Σ_0^B-replacement scheme, but results in [19] suggest that this is unlikely. Instead we show that **VNC**$^1(\mathcal{L}_{TreeRec})$ proves (8) for each $\Sigma_0^B(\mathcal{L}_{TreeRec})$-formula ϕ, by structural induction on ϕ. The induction step, when ϕ is built from simpler formulas from the Boolean operations or bounded number quantification, is straightforward. For example, if $\phi(z)$ is $\exists x \le t\psi(x,z)$, then using the pairing function (10) we have by the induction hypothesis

$$\mathbf{VNC}^1(\mathcal{L}_{TreeRec}) \vdash \exists X \forall x \le t \forall z < y(X(x,z) \leftrightarrow \psi(x,z))$$

Now by Σ_0^B-COMP,

$$\mathbf{V}^0 \vdash \exists X' \le y \forall z < y(X'(z) \leftrightarrow \exists x \le t\, X(x,z))$$

Thus **VNC**$^1(\mathcal{L}_{TreeRec})$ proves comprehension for ϕ.

The base case of the induction is straightforward except for the case of one of the new relation symbols $R^{\phi,\psi}$. Here it suffices to show **VNC**$^1(\mathcal{L}_{TreeRec})$ proves (8) where $\phi(z,\mathbf{x},\mathbf{Y})$ is replaced by $R^{\phi,\psi}(a,i,\mathbf{x},\mathbf{X})$ when z is one of the number variables $a,i,\mathbf{x}$. The case in which z is i follows from (15). Now consider the case in which z is in $\mathbf{x}$. (The case in which z is a is similar.) To simplify notation, we assume x is $\mathbf{x}$. By (16) it suffices to show

$$\mathbf{VNC}^1 \vdash \exists W \forall x < y[W(x) \leftrightarrow \exists Z \le 2a(B^{\phi,\psi}(a,x,\mathbf{X},Z) \wedge Z(i))] \tag{17}$$

The RHS of this defines W in terms of trees Z_x for $x = 0,1,...,y-1$. In order to show that the existence of W follows from the Σ_0^B-$TreeRec$ scheme (13) we collect all of these trees into one large tree U which has them attached to y leaves of the top part of U.

To describe in **VNC**^{1} these tree embeddings we use the fact that the first order theory $\mathbf{I\Delta_0}$, and hence **VNC**^{1}, defines functions such as $|x|$ (the length of x in binary) and $2^{|x|}$ and proves their basic properties (see for example [8,16]).

Tree Z_x is represented in U by the subtree of U rooted at node $root(x) = 2^{|y|} + x$. Note that these y root nodes are consecutive nodes at level $|y|$ in the tree U (where the root of U is at level 0). In general, node i of tree Z_x is at level $level(i) = |i| - 1$ in Z_x and hence at level $level(i) + |y|$ in U. In fact, node i in tree Z_x is represented by node

$$node(i,x) = root(x) \cdot 2^{level(i)} + (i - 2^{level(i)})$$

in U. Note that the leaves of Z_x are at level $|a-1|$ in Z_x, except some may be at level $|a-1|-1$. The leaves of interest in U are the deeper leaves of the embedded trees Z_x, and these have level $|a-1| + |y|$.

The function $node(i, x)$ is injective for pairs i, x such that $1 \leq i$ and $0 \leq x < y$, and its inverses $icomp(j)$ and $xcomp(j)$ are definable in $I\Delta_0$, and $I\Delta_0$ proves for i, x satisfying these conditions, that if $j = node(i, x)$, then $i = icomp(j)$ and $x = xcomp(j)$.

The formulas $\phi(i, x, \mathbf{X})[p, q], \psi(i, x, \mathbf{X})$ used to define the tree Z_x determine $\phi'(j, y, a, \mathbf{X})[p, q], \psi'(j, y, a, \mathbf{X})$ to define the tree U, where

$$\begin{aligned}\phi'(j, y, a, \mathbf{X})[p, q] \equiv\ & icomp(j) < a \wedge \phi(icomp(j),\ xcomp(j), \mathbf{X})[p, q]\ \vee \\ & icomp(j) \geq a \wedge \psi(icomp(j) \dot{-} a,\ xcomp(j), \mathbf{X})\end{aligned}$$

$$\psi'(j, y, a, \mathbf{X}) \equiv \psi(icomp(j + a') \dot{-} a,\ xcomp(j + a'), \mathbf{X})$$

where a' is defined below. By (13) we have

$$\mathbf{VNC}^1 \vdash \exists U \leq 2a'\ B^{\phi', \psi'}(a', y, a, \mathbf{X}, U) \tag{18}$$

where $a' = 2^{|a \dot{-} 1| + |y|}$. The reason for this value of a' is that all leaves of the tree U are at level $|a - 1| + |y|$, as noted above.

Finally $\mathbf{VNC}^1$ proves the existence of W in (17) using Σ_0^B-COMP and the definition

$$W(x) \leftrightarrow U(node(i, x))$$

where U is obtained from (18). In order to prove that W defined in this way satisfies (17), $\mathbf{VNC}^1$ proves each tree Z_x is embedded as claimed in U; that is

$$\begin{aligned}& 0 < i < 2a \wedge x < y \wedge B^{\phi, \psi}(a, x, \mathbf{X}, Z) \wedge B^{\phi', \psi'}(a', y, a, \mathbf{X}, U)\ \supset \\ & \quad (Z(i) \leftrightarrow U(node(i, x)))\end{aligned}$$

This can be done using Σ_0^B-IND on $(2a \dot{-} i)$. □

Recall from Definition 11 that a string function $F(\mathbf{x}, \mathbf{X})$ is in $\mathbf{FNC}^1$ iff it is p-bounded and its bit graph is in $\mathbf{NC}^1$. It is easy to check that a number function $f(\mathbf{x}, \mathbf{X})$ is in $\mathbf{FNC}^1$ iff it satisfies $f(\mathbf{x}, \mathbf{X}) = |F(\mathbf{x}, \mathbf{X})|$ for some string function F in $\mathbf{FNC}^1$.

Let $\mathbf{VNC}^1(\mathbf{FNC}^1)$ be the extension of of $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$ obtained by adding to the language $\mathcal{L}_{TreeRec}$, for every $\Sigma_0^B(\mathcal{L}_{TreeRec})$ formula $\phi(i, \mathbf{x}, \mathbf{X})$ and term $t(\mathbf{x}, \mathbf{X})$ of $\mathcal{L}_A^2$, a function symbol F and its bit-graph defining axiom (12). By Lemma 12 and the definition of $\mathbf{FNC}^1$ it is clear that the functions symbols in $\mathbf{VNC}^1(\mathbf{FNC}^1)$ represent precisely the functions in $\mathbf{FNC}^1$.

Theorem 18. $\mathbf{VNC}^1(\mathbf{FNC}^1)$ *is a conservative extension of* $\mathbf{VNC}^1$. *Every* $\Sigma_1^B(\mathbf{FNC}^1)$ *theorem of* $\mathbf{VNC}^1(\mathbf{FNC}^1)$ *is witnessed in* $\mathbf{VNC}^1(\mathbf{FNC}^1)$. *The* Σ_1^B*-definable functions in* $\mathbf{VNC}^1$ *are precisely those in* $\mathbf{FNC}^1$.

Proof. To prove the first sentence it suffices to show that $\mathbf{VNC}^1(\mathbf{FNC}^1)$ is conservative over $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$. To do this it suffices to show that each new function F introduced by its bit-graph defining axiom (12) is $\Sigma_1^B(\mathcal{L}_{TreeRec})$-definable in $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$. This in turn follows from Lemma 13.

The second sentence follows from Theorem 17. The third sentence follows from the first two, and the fact that every relation $R^{\phi, \psi}$ can be Δ_1^B-defined in $\mathbf{VNC}^1$. □

7.2. **VNC**^{1} *is finitely axiomatizable*

Theorem 19. VNC^1 *is finitely axiomatizable.*

Proof. (Sketch) By a result in [18], $\mathbf{V}^0$ is finitely axiomatizable. Hence to show that **VNC**^{1} is finitely axiomatizable it suffices to show that one particular instance of the Σ_0^B-$TreeRec$ scheme (13) implies them all. That instance is based on Clote's function $tree(x)$, which evaluates an and-or tree whose inputs are the bits of x. This function is complete for **NC**^{1}, and is used in [15] to define an algebra for **NC**^{1} functions and in [14] to define the equational theory ALV for **NC**^{1}.

We call TE the particular instance of (13) with $\phi \equiv \phi^{\mathbf{TE}}$ and $\psi \equiv \psi^{\mathbf{TE}}$, where $\psi^{\mathbf{TE}}(i, X) \equiv X(i)$ and

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

where $parity(x)$ holds iff x is odd. Then

$$\mathbf{TE} \equiv \forall a \forall X \exists Z \leq 2a B^{\phi^{\mathbf{TE}}, \psi^{\mathbf{TE}}}(a, X, Z)$$

We claim that $\mathbf{V}^0$ can prove any instance of (13) from **TE**:

$$\mathbf{V}^0 \vdash \mathbf{TE} \supset \exists Z \leq 2a B^{\phi,\psi}(a, \mathbf{x}, \mathbf{X}, Z)$$

The idea is to use **TE** to construct a large tree Z' from which the required tree Z can be extracted using Σ_0^B-COMP. There are 16 binary Boolean functions $\beta_0(p, q), ..., \beta_{15}(p, q)$. Each β_k can be computed by an and-or tree of depth 3 with inputs among $0, 1, p, q, \neg p, \neg q$. These 16 trees can be combined into an and-or tree T of depth 12 with inputs $v_0, ..., v_{15}, 0, 1, p, q, \neg p, \neg q$ whose output is $\beta_k(p, q)$ when the inputs satisfy $v_k = 1$ and $v_i = 0$ for $i \neq k$.

The tree Z' has two copies T_i and T_i' of T for each internal node i in Z. These are arranged so that the output of T_i is $Z(i)$ and the output of T_i' is $\neg Z(i)$. The inputs $p, q, \neg p, \neg q$ of both T_i and T_i' are attached to the outputs of $T_{2i}, T_{2i+1}, T_{2i}', T_{2i+1}'$, respectively. The other inputs $v_0, ..., v_{15}, 0, 1$ are channeled up from the leaves of Z'. These leaves take values determined by X in $\psi^{\mathbf{TE}}(i, X)$, and X can be defined appropriately from ϕ, ψ using Σ_0^B-COMP. $\square$

Details of the above argument appear in [29], and a similar construction for the theory **AID** (see below) appears in section 9 of [1]. In fact [1] shows that there are Σ_1^b-faithful interpretations between **AID** + Σ_0^b-CA and $QALV$, a quantified version of Clote's equational theory ALV. Thus by the RSUV results below there are similar interpretations between **VNC**^{1} and $QALV$.

7.3. **AID** *and RSUV isomorphism*

AID [1] is a first-order theory of bounded arithmetic defined over a base language $\mathcal{L}_{BA}$ consisting of the function symbols $0, 1, +, \lfloor x/2 \rfloor, |x|, x\#y, x{\cdot}2^{|y|}, x \dot{-} y, x[i, j)$. The Σ_0^b formulas of **AID** are the sharply bounded formulas in this language. It is important to note that $\mathcal{L}_{BA}$ does *not* contain integer multiplication $x \cdot y$, but $i \cdot j$ is Σ_0^b definable for small values $i, j \leq |x|$, given x.

The axioms for **AID** iinclude a set BASIC of Σ_0^b axioms defining the properties of the functions in $\mathcal{L}_{BA}$.

In addition to $\mathcal{L}_{BA}$, the language $\mathcal{L}_{AID}$ of **AID** iincludes an $n+1$-ary predicate symbol $A^{\ell,B,\overline{D},I}$ for each $\ell, B, \overline{D}, I$, where ℓ is a linear form in $\|x_1\|, ..., \|x_n\|$, and $B(\overline{x},p), D_1(\overline{x},p), ..., D_m(\overline{x},p)$ are Σ_0^b formulas and I is a propositional formula in the atoms $(d_1, ..., d_m, p_0, p_1)$. These predicate symbols $A^{\ell,B,\overline{D},I}$ are intended to represent the **Alogtime** predicates. Each such symbol has the following defining axioms, which define it inductively over a binary tree.

(A.0) $A(\overline{x},p) \supset 0 \neq |p| \leq \ell\|\overline{x}\|$,

(A.1) $0 \neq |p| = \ell\|\overline{x}\| \supset [A(\overline{x},p) \leftrightarrow B(\overline{x},p)]$,

(A.2) $0 \neq |p| < \ell\|\overline{x}\| \supset [A(\overline{x},p) \leftrightarrow I(\overline{D}(\overline{x},p), A(\overline{x},2p), A(\overline{x},2p+1))]$.

Finally **AID** contains the $\Sigma_0^b(L_{AID})$-LIND scheme

$$A(0) \wedge \forall y < |x|(A(y) \supset A(y+1)) \supset A(|x|)$$

where L_{AID} is the language of **AID**.

AID does *not* contain the following comprehension scheme:

$$\Sigma_0^b\text{-CA:} \qquad \exists |y| \leq p|\overline{x}| \forall i < p|\overline{x}|(bit(i,y) \leftrightarrow B(i,\overline{x})) \tag{19}$$

where $p|\overline{x}|$ is a polynomial and $bit(i,y)$ holds iff bit i in the binary notation for y is 1. However **AID** + Σ_0^b-CA is Σ_0^b-conservative over **AID** (Lemma 8.1 in [1]).

Theorem 20. $\mathbf{VNC}^1$ *is RSUV isomorphic to* **AID** + Σ_0^b*-CA.*

Proof. Let us abbreviate **AID** + Σ_0^b-CA by **AID+**. In the present context, an "RSUV isomorphism" [24,33,36,26] is a bijection between (isomorphism types of) models of **AID+** and models of $\mathbf{VNC}^1$. Each model $\mathcal{M}_1$ of **AID+** determines a model $\mathcal{M}_2$ of $\mathbf{VNC}^1$ 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 $\mathbf{VNC}^1$ determines a model $\mathcal{M}_1$ of **AID+** 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 $\mathcal{M}_2'$.

If N and S are the number and string universes for a model of $\mathbf{VNC}^1$ (or of $\mathbf{V}^0$) then we may assume by the extensionality axiom SE that each element of S is a subset of N. Further there is a natural injection of N into S, where an element x of N is sent to the set $X = \{i \mid bit(i,x)\}$. The predicate bit is Σ_0^B-definable in $\mathbf{V}^0$ because it has a bounded definition in $\mathrm{I}\Delta_0$ [8,16]. Thus the set X exists in S by Σ_0^B-COMP, and further $|X| = |x|$, where $|x|$ is the length of the binary notation of x, which is definable in $\mathbf{V}^0$. The image of N in S corresponds to $\log(M_1)$ in a model $\mathcal{M}_1$ of **AID+**, since by Σ_0^B-COMP, N is precisely the set of lengths of strings in S.

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be corresponding models of AID+ and $\mathbf{VNC}^1$ as described above. Then we have a bijection between the universe M_1 of $\mathcal{M}_1$ and the string universe M_2 of $\mathcal{M}_2$, whose restriction gives us a bijection between $\log(M_1)$ and the number part $num(M_2)$ of $\mathcal{M}_2$. This bijection sends every function and predicate of $\mathcal{M}_1$ to a corresponding function or predicate in $\mathcal{M}_2$, and *vice versa*. To complete the RSUV isomorphism we must show that each function and relation in the language of one model can be defined in the other model in such a way that the axioms in the first theory are satisfied.

Let $\mathcal{M}_1$ be a model of **AID+** with universe M_1. Let $\mathcal{M}_2$ be the corresponding model of $\mathbf{VNC}^1$ as described above, so the string universe is M_1 and the number universe is $\log(M_1)$. To complete the definition of $\mathcal{M}_2$ we need to define the functions $0, 1, +, \cdot, |X|, \in$ and the relation $\leq$. We define $0, 1$ and $+$ to be the the restrictions to $\log(M_1)$ of these operations in $\mathcal{M}_1$. The function $\cdot$ is not in the language of **AID** but it can be defined suitably in **AID** for "small numbers" (i.e. numbers of the form $|u|$) [1]. We define $|X|$ according to the binary length function $|u|$ in **AID**, and we define $\in$ by $|u| \in v$ iff $bit(|u|, v)$ holds in $\mathcal{M}_1$. We define $\leq$ in $\mathcal{M}_2$ by restricting $\leq$ in $\mathcal{M}_1$ to $\log(M_1)$.

It is easy to check that $\mathcal{M}_2$ satisfies all of the axioms of $\mathbf{V}^0$, because all axioms correspond to easy theorems of **AID+**. In particular, Σ_0^B-COMP corresponds to Σ_0^b-CA. We argue below that the Σ_0^B-$TreeRec$ scheme holds in $\mathcal{M}_2$.

Conversely, given a model $\mathcal{M}_2$ of $\mathbf{VNC}^1$ we describe below how each of the functions and predicates in the language $\mathcal{L}_{\mathbf{AID}}$ can be interpreted in $\mathcal{M}_2$ so that these interpretations give definitions for them in the corresponding model $\mathcal{M}_1$, and these definitions will satisfy the axioms of **AID+**.

7.4. Syntactic Interpretations between **AID+** *and* $\mathbf{VNC}^1$

It is not hard to give Σ_0^B definitions in $\mathbf{VNC}^1$ for each of the functions $0, 1, +, \lfloor x/2 \rfloor$, $|x|, x\#y, x \cdot 2^{|y|}, x \dot{-} y, x[i, j)$ of **AID**, because each of these is an $\mathbf{AC^0}$ function on binary numbers. Then $\mathbf{V}^0$ proves the BASIC axioms of **AID** involving these functions. Further these Σ_0^B definitions allow us to translate every Σ_0^b formula $C(\overline{x})$ of **AID** to an equivalent Σ_0^B formula $C^\sharp(\mathbf{X})$ of $\mathbf{V}^0$ by translating every variable x of **AID** to a corresponding string variable X of $\mathbf{V}^0$. Note that the Σ_0^b-CA axioms translate to formulas which follow from Σ_0^B-COMP axioms of $\mathbf{V}^0$.

In the other direction, using the interpretations discussed above of $0, 1, +, \cdot, |X|, \in, \leq$, every Σ_0^B formula $\psi(\mathbf{x}, \mathbf{Y})$ can be translated to an equivalent Σ_0^b formula $\psi^\flat(\overline{x}, \overline{y})$ (the $\sharp$ and $\flat$ notation is from [33].)

In order to show that the Σ_0^B-$TreeRec$ scheme (13) holds in $\mathcal{M}_2$ we use the fact that the predicates $A^{\ell,B,\overline{D},I}$ are defined in $\mathcal{M}_1$. Since $\mathcal{M}_2$ satisfies Σ_0^B-COMP, it suffices to define suitable $\ell, B, \overline{D}, I$ such that

$$\forall i < 2a - 1 \; (Z(i) \leftrightarrow A^{\ell,B,\overline{D},I}(b, i))$$

where a in $\mathcal{M}_2$ corresponds to $|b|$ in $\mathcal{M}_1$.

Referring to (13), we have for the base case of the tree recursion

$$Z(i) \leftrightarrow \psi(i - a), a \leq i \leq 2a - 1$$

Since the predicates A in **AID** are defined by tree recursion only for a complete binary tree, we need to fill out the tree so that its leaves are numbered only for those p such that $|p| = \ell||\overline{x}||$. We choose

$$\ell||b|| = ||b|| + 1 = |a| + 1$$

so $|p| = \ell$ iff $2^{|a|} \leq p \leq 2^{|a|+1} - 1$. Then we want the leaf value $B(b, y, p)$ to be $\psi(p - a)$ if $a \leq p \leq 2a - 1$ and $B(b, y, p)$ is $\psi(i - a)$ if $2a \leq p$, where $a = |b|$ and i is the parent node of node p. Thus we define the Σ_0^b formula B by

$$B(b, y, p) \equiv \begin{cases} \psi(p - |b|) & \text{if } p \leq 2|b| - 1 \\ \psi(\lfloor p/2 \rfloor - |b|) & \text{if } 2|b| \leq p \end{cases}$$

Each internal node p of the recursion tree for A computes its value from the values p_0, p_1 of its two children, using one of the 16 possible binary truth functions. We define d_{jk}, where $j, k \in \{0, 1\}$, to be the value of this function on inputs j, k, where 0 = False and 1 = True. Then we define the propositional formula $I(d_{00}, d_{01}, d_{10}, d_{11}, p_0, p_1)$ to specify this function. Thus

$$\begin{aligned} &I(d_{00}, d_{01}, d_{10}, d_{11}, p_0, p_1) \equiv \\ &d_{00}\bar{p}_0\bar{p}_1 \vee d_{01}\bar{p}_0 p_1 \vee d_{10} p_0 \bar{p}_1 \vee d_{11} p_0 p_1 \end{aligned}$$

We define the Σ_0^b formulas D_{jk} to specify d_{jk}, which according to (13) depend on ϕ as follows:

$$D_{jk}(b, y, p) \equiv \begin{cases} \phi(p)[j, k] & \text{if } p < |b| \\ j & \text{if } |b| \leq p \end{cases}$$

The reason for the case $|b| \leq p$ is that then p is not an internal node of the recursion tree of (13), so we want p to take on the value of one of its children.

To complete the interpretation of **AID+** in $\mathbf{VNC}^1$ we argue that each predicate $A^{\ell, B, \overline{D}, I}$ is definable in $\mathbf{VNC}^1$. Thus given $\ell, B, \overline{D}, I$ let $\mathbf{X}$ interpret in $\mathbf{VNC}^1$ the variables $\overline{x}$ in **AID**, and using the translation $C \rightsquigarrow C^\sharp$ of Σ_0^b to Σ_0^B formulas we define

$$\begin{aligned} a &= 2^{\ell \|\bar{X}\| - 1} \\ \psi(i, \mathbf{X}) &\equiv B^\sharp(\mathbf{X}, i + a) \\ \phi(i, \mathbf{X})[p, q] &\equiv I(\overline{D}^\sharp(\mathbf{X}, i), p, q) \end{aligned}$$

Then the translations of $A^{\ell, B, \overline{D}, I}(\overline{x}, p)$ is $R^{\phi, \psi}(2^{\ell \|\bar{X}\| - 1}, p, \mathbf{X})$ (see (16)), and $\mathbf{VNC}^1$ proves the translations of the axioms A.0, A.1, A.2.

Finally we need to verify that the translations of the $\Sigma_0^b(L_{AID})$-LIND axioms of **AID** are provable in $\mathbf{VNC}^1$. The $\Sigma_0^b(L_{AID})$ formulas translate into $\Sigma_0^B(\mathcal{L}_{TreeRec})$ formulas. By Lemma 13, $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$ proves the comprehension scheme for these formulas and hence by (9) it proves $\mathbf{VNC}^1(\mathcal{L}_{TreeRec})$-IND. This suffices. □ (Theorem 20)

By the proof of the above theorem we can obtain a strengthening of Lemma 7.2 of [1]. The isomorphism given in the proof actually describes an isomorphism between the theory Σ_0^b-RD described in [1] and $\mathbf{VNC}^1$. Therefore we obtain the following result, not mentioned in [1].

Theorem 21. *Σ_0^b-RD is equivalent to* **AID** + *Σ_0^B-CA.*

8. Propositional translations

Here we give a complete definition of the translation of bounded formulas over $\mathcal{L}_A^2$ to quantified propositional formulas. This translation is "RSUV equivalent" to the translation of the first-order language of bounded arithmetic (see [26] sec 9.2), which is the one used by [1] to translate **AID**, but our second-order setting allows a much simpler translation [16].

We translate each bounded predicate calculus formula $\phi(\mathbf{X})$ over $\mathcal{L}_A^2$ to a polynomial size family $\|\phi(\mathbf{X})\|[\mathbf{n}]$ 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 $\phi(\mathbf{X})$ 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 $\|\phi(\mathbf{X})\|[\mathbf{n}]$ and tuples of strings $\mathbf{X}$, with $|X_i| = n_i$, satisfying $\phi(\mathbf{X})$.

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

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

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

For the induction step, $\phi(\mathbf{X})$ 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

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

For the case of bounded number quantifiers, we define

$$\|\exists y \leq t(|\mathbf{X}|)\psi(y, \mathbf{X})\|[\mathbf{n}] = \bigvee_{i=0}^{m} \|\psi(i, \mathbf{X})\|[\mathbf{n}]$$

$$\|\forall y \leq t(|\mathbf{X}|)\psi(y, \mathbf{X})\|[\mathbf{n}] = \bigwedge_{i=0}^{m} \|\psi(i, \mathbf{X})\|[\mathbf{n}]$$

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

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

$$\begin{aligned}
&\|\exists Y \le t(|\mathbf{X}|)\psi(Y,\mathbf{X})\|[\mathbf{n}] = \\
&\qquad \exists p_0^Y \ldots \exists p_{m-2}^Y \bigvee_{i=0}^{m} \|\psi(Y,\mathbf{X})\|[i,\mathbf{n}] \\
&\|\forall Y \le t(|\mathbf{X}|)\psi(Y,\mathbf{X})\|[\mathbf{n}] = \\
&\qquad \forall p_0^Y \ldots \forall p_{m-2}^Y \bigwedge_{i=0}^{m} \|\psi(Y,\mathbf{X})\|[i,\mathbf{n}]
\end{aligned}$$

where again $m = val(t(\mathbf{n}))$. (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 $\|\phi(\mathbf{X})\|[\mathbf{n}]$ of $\phi(\mathbf{X})$. 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 $\phi(\mathbf{x},\mathbf{X})$ over the language $\mathcal{L}_A^2$ there is a polynomial $p(\mathbf{x},\mathbf{y})$ such that the QPC formula $\|\phi(\mathbf{r},\mathbf{X})\|[\mathbf{n}]$ is bounded in size by $p(\mathbf{r},\mathbf{n})$. Further, if $\phi(\mathbf{x},\mathbf{X})$ is Σ_0^B, then the $\wedge$ - $\vee$ alternation depth of $\|\phi(\mathbf{r},\mathbf{X})\|[\mathbf{n}]$ is bounded, independent of $\mathbf{r},\mathbf{n}$.

This translation allows us to state a number of results, which can be inferred from the literature [27,26,16,17], connecting a theory T over $\mathcal{L}_A^2$ with a corresponding QPC proof system. For example, a Σ_0^B-theorem of $\mathbf{V}^0$ translates to a tautology family with polynomial size bounded-depth Frege proofs. Second-order analogs of the G_i simulation theorems for S_2^i and T_2^i are presented as Theorem 23 below.

The next result shows that $\mathbf{VNC}^1$ proofs of bounded formulas translate into polynomial size families of G_0^* proofs. This is analogous to Arai's [1] theorem showing that **AID** proofs of Σ_0^b formulas translate into polynomial size families of Frege proofs. Our result is more general, because it applies to all bounded theorems and not just those in Σ_0^B, and simpler, because of our second-order setting.

Theorem 22. $\mathbf{VNC}^1$ Simulation: *If $\phi(\mathbf{x},\mathbf{X})$ is a bounded theorem of $\mathbf{VNC}^1$, then the family $\|\phi(\mathbf{r},\mathbf{X})\|[\mathbf{n}]$ has G_0^* proofs of size polynomial in $\mathbf{r},\mathbf{n}$, and can be computed by an $\mathbf{NC}^1$ function of $\mathbf{r},\mathbf{n}$.*

Proof. Let π be an anchored treelike LK proof of a bounded formula $\phi(\mathbf{x},\mathbf{X})$ from the axioms of $\mathbf{VNC}^1$. Thus all cut formulas of π are substitution instances of $\mathbf{VNC}^1$ axioms, and hence are Σ_1^B, and therefore all formulas in π are bounded. Given $(\mathbf{r},\mathbf{n})$, each sequent $S(\mathbf{x},\mathbf{X})$ of π can be transformed to a sequent $S'(\mathbf{r},\mathbf{n})$ of QPC formulas by transforming each formula $\psi(\mathbf{x},\mathbf{X})$ to $\|\psi(\mathbf{r},\mathbf{X})\|[\mathbf{n}]$. Thus by induction on the length of π, it is straightforward to find a G_1^* derivation $\pi[\mathbf{r},\mathbf{n}]$ of $\|\phi(\mathbf{r},\mathbf{X})\|[\mathbf{n}]$, where each nonlogical axiom and each non-Σ_0^q cut formula of $\pi[\mathbf{r},\mathbf{n}]$ is a translation D of some substitution instance of an axiom of $\mathbf{VNC}^1$. (See [16], pp 103-105, for details.) It remains to transform $\pi[\mathbf{r},\mathbf{n}]$ to a G_0^* proof with no nonlogical axioms.

All axioms of $\mathbf{VNC}^1$ are Σ_0^B except Σ_0^B-COMP and Σ_0^B-$TreeRec$. Each of the Σ_0^B axioms either translates to T, or translates to a valid Σ_0^q-formula with a trivial G_0^* proof. Each of the Σ_1^B axioms of $\mathbf{VNC}^1$ has the form $\exists Y \le \mathbf{t}\psi(\mathbf{x},\mathbf{X},Y)$, where ψ is Σ_0^B. Further, given $\mathbf{r},\mathbf{n}$, it is easy to find quantifier-free formulas $C_0,\ldots,C_{m-2}$ witnessing the existential quantifiers $\exists x_0^Y \ldots \exists x_{m-2}^Y$ in its translation

$$D \;\equiv\; \|\exists Y \le t\psi(\mathbf{r},\mathbf{X},Y)\|[\mathbf{n}] \tag{20}$$

where $m = val(t)$. In fact, if these existential quantifiers are removed from D and each variable x_i^Y is replaced by C_i, the result is a valid Σ_0^q formula D' with a G_0^* proof of size polynomial in $\mathbf{r}, \mathbf{n}$.

Now consider an uppermost instance of the cut rule in $\pi[\mathbf{r}, \mathbf{n}]$, with cut formula D from (20). We change this instance to an instance in which the cut formula is D' instead of D, but the conclusion is the same. The left hypothesis of the original instance has D in the consequent: just replace D by D' after deriving D' with a G_0^* proof. The right hypothesis has D in the antecedent: modify the derivation of this sequent by replacing every eigenvariable p_i^Y in an exists-left rule by C_i throughout the derivation, and remove all $\exists$-left rules used to derive D. Thus we convert

$$\frac{\begin{array}{c}\vdots\\ \Gamma \to \Delta, D\end{array} \qquad \begin{array}{c}\vdots\\ D, \Gamma \to \Delta\end{array}}{\Gamma \to \Delta}$$

to

$$\frac{\dfrac{\begin{array}{c}\vdots\\ \to D'\end{array}}{\Gamma \to \Delta, D'} \qquad \begin{array}{c}\vdots\\ D', \Gamma \to \Delta\end{array}}{\Gamma \to \Delta}$$

The result is a G_0^* derivation of the same sequent.

Continue replacing each Σ_1^q cut formula in the proof $\pi[\mathbf{r}, \mathbf{n}]$ by a Σ_0^q cut formula, in the same way. $\square$

Since a G_0 proof of a quantifier-free formula is a Frege proof, we obtain

Corollary 1. *If $\phi(\mathbf{x}, \mathbf{X})$ is a Σ_0^B-theorem of* **VNC**^{1}*, then the family $\|\phi(\mathbf{r}, \mathbf{X})[\mathbf{n}]$ has Frege proofs of size polynomial in $\mathbf{r}, \mathbf{n}$, and these can be computed in* **FNC**^{1}.

We point out that Theorem 22 together with the G_0 witnessing theorem 9 give an alternative proof that the Σ_1^B theorems of **VNC**^{1} can be witnessed by **FNC**^{1} functions. Thus given values $\mathbf{r}, \mathbf{P}, \mathbf{n}$ for the free variables $\mathbf{x}, \mathbf{X}$ (where $\mathbf{n}$ gives the lengths of the strings $\mathbf{P}$) in a theorem $\exists Y \phi(\mathbf{x}, \mathbf{X}, Y)$ of **VNC**^{1} we compute a witnessing value for Y by first computing a G_0^* proof of $\|\exists Y \phi(\mathbf{r}, \mathbf{X})\|[\mathbf{n}]$ and then use G_0 witnessing to compute the bits $p_0^Y, p_1^Y, \ldots$ of Y, after using $\mathbf{P}$ to evaluate the propositional variables for $\mathbf{X}$.

The simulation theorem 22 has versions for many theories over the language $\mathcal{L}_A^2$ [17], and in particular the first-order simulation theorems for the S_2^i and T_2^i hierarchies [27] have nice second-order settings. The theory **V**^{0} (section 6.3) generalizes to the theory **V**^{i} over the same language replacing the comprehension scheme Σ_0^B-COMP by Σ_i^B-COMP. For $i \geq 1$, **V**^{i} is essentially the theory $\mathbf{V}_1^i$, and is RSUV isomorphic to S_2^i. Similarly the theory **TV**^{i} is obtained from **V**^{0} by adding the Σ_i^B-String-IND scheme, which provides string induction for Σ_i^B formulas when strings X are treated as binary numbers. Then **TV**^{0} is a second-order version of the polynomial time theory $\mathbf{PV}_1$, and for $i \geq 1$, **TV**^{i} is RSUV isomorphic to T_2^i.

The following result gives a second-order setting to the Krajíček -Pudlák [27] result showing G_i simulates T_2^i, and to the Krajíček result [26] showing G_i^* simulates S_2^i. Our modified definitions of G_i and G_i^* allow us to state the result for arbitrary bounded theorems of $\mathbf{TV}^i$ and $\mathbf{V}^i$, as opposed to just Σ_i^B theorems.

Theorem 23. *For $i \geq 1$ if $\phi(\mathbf{x}, \mathbf{X})$ is a bounded theorem of $\mathbf{V}^i$, then the family $\|\phi(\mathbf{r}, \mathbf{X})\|[\mathbf{n}]$ has G_i^* proofs which can be computed in time polynomial in $\mathbf{r}, \mathbf{n}$. The same is true for $\mathbf{TV}^i$ and G_i.*

Proof. (sketch) The theories $\mathbf{V}^i$ and $\mathbf{TV}^i$ can be formulated as LK systems with the 2-$BASIC$ axioms and Σ_0^B-COMP formulated as a Σ_1^B scheme, together with suitable Σ_i^B induction rules. Then all cut formulas in an anchored (i.e. free-cut-free) proof of a bounded formula are Σ_i^B. In the case of $\mathbf{V}^i$ the G_i^* proofs are formed as in the proof of Theorem 22. However each use of the Σ_i^B induction rule must be translated into a polynomial size sequence of cuts in the G_i^* proof, with Σ_i^q cut formulas. In the case of $\mathbf{TV}^i$, a straightforward translation of the induction rule would result in exponentially many cuts, so instead we use a doubling chain of implications whose intuitive meaning is $\|\psi(X)\| \rightarrow \|\psi(X + 2^i)\|$, where $+$ is binary addition. This G_i proof is not treelike, and uses the fact that for $i \geq 1$ a substitution rule for Σ_i^q formulas can be added to G_i with only a polynomial increase in power. For more details, see [16,26] in the case $\mathbf{V}^i$, and [27,26] in the case $\mathbf{TV}^i$. $\square$

9. Concluding Remarks and Open Problems

Let $i, j \geq 1$. It is known that if $j \leq i$ the j-reflection principles for G_i and G_i^* (soundness of G_i and G_i^* for proving Σ_j^q theorems) are provable in T_2^i and S_2^i, respectively [27,26], and that S_2^1 plus the j-reflection principle for G_i or G_i^* axiomatize the Σ_j^b-consequences of the corresponding theory [26]. Now that we have modified the definitions of G_i and G_i^* it is natural to ask whether these same relationships hold for $j > i$. As we remarked after Theorem 7, the provability of the j-reflection principles in the corresponding theories for such j imply upper bounds on the complexity of the Σ_j^q-witnessing in the same way the similar upper bounds are obtained in the proof.

For $i = 0$, the 0-reflection principle for G_0 is provable in $\mathbf{VNC}^1$. This is true because Arai [1] shows that **AID** proves the soundness of Frege systems, and we have shown that **AID** is RSUV isomorphic to $\mathbf{VNC}^1$. It seems likely that the 1-reflection principle for G_0 is also provable in $\mathbf{VNC}^1$, since our proof of Theorem 9 (Σ_1^q-witnessing for G_0) should be formalizable in $\mathbf{VNC}^1$. Note that this will require a formalization in $\mathbf{VNC}^1$ of Theorem 8 (the correctness of the π-witness formulas), which seems feasible since, as we pointed out at the end of Section 4.1, it is $\mathbf{TC}^0$ to construct short PK-proofs of the correctness of π-witness formulas.

Also it is plausible that Theorem 22 (Simulation of $\mathbf{VNC}^1$ by G_0^*) is formalizable in $\mathbf{VNC}^1$ in the following way: if $\phi(\mathbf{x}, \mathbf{X})$ is a bounded theorem of $\mathbf{VNC}^1$, then $\mathbf{VNC}^1$ proves that the family $||\phi(\mathbf{x}, \mathbf{X})||[\mathbf{n}]$ has polynomial-size G_0^*-proofs.

By Theorem 12 the Σ_1^q witnessing problem for G_0 and G_0^* are complete for $\mathbf{NC}^1$. For $j \geq 2$, the complexity of the Σ_j^q witnessing problems for G_0 and G_0^* are open, and are related to the (unknown) complexity of witnessing Σ_j^B theorems of **VNC**^{1}.

Acknowledgements. We wish to thank the anonymous reviewer for helpful comments and suggestions on the draft of this paper.

References

1. T. Arai. A bounded arithmetic AID for Frege systems. *Annals of Pure and Applied Logic*, 103:155–199, 2000.
2. D. A. M. Barrington, N. Immerman, and H. Straubing. On Uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.
3. S. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.
4. S. Buss. The Boolean formula value problem is in ALOGTIME. *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*, pages 123–131, 1987.
5. 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.
6. S. Buss. Propositional consistency proofs. *Annals of Pure and Applied Logic*, 52:3–29, 1991.
7. S. Buss. Algorithms for Boolean formula evaluation and for tree-contraction. In P. Clote and J. Krajicek, editors, *Proof Theory, Complexity, and Arithmetic*, pages 95–115. Oxford University Press, 1993.
8. 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/`.
9. 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/`.
10. S. Buss, 2003. Personnal communication.
11. S. Buss and P. Clote. Cutting planes, connectivity, and threshold logic. *Archive for Mathematical Logic*, 35:33–62, 1996.
12. S. Buss and J. Krajicek. An application of Boolean complexity to separation problems in bounded arithmetic. *The Proceedings of the London Mathematical Society*, 60(3):1–21, 1994.
13. M. Chiari and J. Krajicek. Witnessing functions in bounded arithmetic and search problems. *The Journal of Symbolic Logic*, 63:1095–1115, 1998.
14. P. Clote. $ALOGTIME$ and a conjecture of S.A. Cook. *Ann. Math. Art. Intell.*, 6:57–106, 1990. extended abstract in Proc. 13th IEEE Symposium on Logic in Computer Science, 1990.
15. P. Clote. Sequential, machine-independent characterizations of the parallel complexity classes $AlogTIME, AC^k, NC^k$ and NC. In S. Buss and P. Scott, editors, *Feasible Mathematics*, pages 49–69. Birkhauser, 1990.
16. S. Cook. Csc 2429 course notes: Proof complexity and bounded arithmetic, 2002. Available from the web at `www.cs.toronto.edu/~sacook/csc2429h/`.
17. S. Cook. Theories for complexity classes and their propositional translations. *submitted*, pages 1–36, 2004.
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 N. Thapen. The strength of replacement in weak arithmetic. *manuscript*, pages 1–19, 2003.
20. S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1977.

21. N. Immerman. *Descriptive Complexity*. Springer, 1999.
22. D. S. Johnson. A catalog of complexity classes. In J. van Leewen, editor, *Handbook of Theoretical Computer Science*, pages 67–161. Elsevier Science Publishers, 1990.
23. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.
24. J. Krajíček. Exponentiation and second-order bounded arithmetic. *Annals of Pure and Applied Logic*, 48:261–276, 1990.
25. J. Krajicek. Fragments of Bounded Arithmetic and Bounded Query Classes. *Transactions of the American Mathematical Society*, 338(2):587–598, 1993.
26. J. Krajíček. *Bounded Arithmetic, Propositional Logic and Computational Complexity*. Cambridge University Press, 1995.
27. J. Krajíček and P. Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift f. Mathematische Logik u. Grundlagen d. Mathematik*, 36:29–46, 1990.
28. T. Morioka. Search problems, proof complexity, and quantified propositional calculus. PhD thesis in prepatation, the Department of Computer Science, the University of Toronto, 2004.
29. P. Nguyen. Proving that $\mathbf{VNC}^1$ is finitely axiomatizable. unpublished note, 2004.
30. T. Pitassi. Using hardness to prove Frege lower bounds, 2002. A seminar at the Fields Institute for Research in Mathematical Sciences, Toronto, Canada.
31. C. Pollett. A propositional proof systems for R^i_2. In P. Beame and S. Buss, editors, *Proof Complexity and Feasible Arithmetics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 253–278. The American Mathematical Society, 1997.
32. C. Pollett. Structure and Definability in General Bounded Arithmetic Theories. *Annals of Pure and Applied Logic*, 100:189–245, 1999.
33. A. A. Razborov. An equivalence between second order bounded domain bounded arithmetic and first order bounded arithmetic. In P. Clote and J. Krajicek, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 247–77. Oxford University Press, 1993.
34. A. Skelley. Personal communication., 2002.
35. G. Takeuti. *Proof Theory*. Elsevier Science Publisers, second edition, 1987.
36. G. Takeuti. RSUV isomorphism. In P. Clote and J. Krajicek, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–86. Oxford University Press, 1993.
37. D. Zambella. Notes on polynomially bounded arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.