A second-order system for polynomial-time reasoning based on Grädel's theorem*

Stephen Cook and Antonina Kolokolova †
Dept. of Computer Science, University of Toronto
{sacook,kol}@cs.toronto.edu
http://www.cs.toronto.edu/~{sacook,kol}

January 8, 2001

Abstract

We introduce a second-order system V_1-Horn of bounded arithmetic formalizing polynomial-time reasoning, based on Grädel's [9] second-order Horn characterization of P. Our system has comprehension over P predicates (defined by Grädel's second-order Horn formulas) , and only finitely many function symbols. Other systems of polynomial-time reasoning either allow induction on NP predicates (such as Buss's S_2^1 or the second-order V_1^1), and hence are more powerful than our system (assuming the polynomial hierarchy does not collapse), or use Cobham's theorem to introduce function symbols for all polynomial-time functions (such as Cook's PV and Zambella's P-def). We prove that our system is equivalent to QPV and Zambella's P-def. Using our techniques, we also show that V_1-Horn is finitely axiomatizable, and, as a corollary, that the class of $\forall\Sigma_1^b$ consequences of S_2^1 is finitely axiomatizable as well, thus answering an open question.

1 Introduction

1.1 Bounded Arithmetic

The first theory that was explicitly designed in order for all proofs to be feasibly constructible (i.e., constructible in polynomial time) was the equational theory PV, proposed by Cook in 1975 [3]. There, Cobham's characterization of polynomial-time was used to construct polynomial-time functions. One motivation for PV was its close relation with Extended Frege proof systems for the propositional calculus: theorems of PV give rise to families of tautologies with polynomial-length proofs.

A major work establishing the relation between complexity theory and bounded arithmetic was the 1985 PhD thesis of S.Buss [1], where the first-order theories S_2^i and various second-order theories were developed that characterize the levels of the polynomial-time hierarchy, PSPACE and EXPTIME. The most important of them is the first-order theory S_2^1, consisting of a set of 32 axioms and an induction-on-notation scheme over Σ_1^b (NP) formulas. Buss proves that a function is Σ_1^b- definable in S_2^1 iff it is polynomial-time. He also shows that S_2^1 is $\forall\Sigma_1^b$ conservative over QPV (a quantified

*The full version of this paper is available on the authors' webpages

†Contact author. Phone: (416) 978-6119, Fax: (416) 978-1931, Address: Dept. of Computer Science, University of Toronto, 10 King's College Road, Toronto, ON M5S 3G4, Canada

version of Cook's PV); however general conservativity of S_2^1 over QPV would imply the collapse of the polynomial hierarchy to P/poly [2, 13, 21]. Razborov and, at the same time, Takeuti [16, 19] introduce a general method for showing the equivalence between certain first-order and second-order theories, which can be used to show that the second-order theory V_1^1 is equivalent in power to S_2^1 [16]. Finally, in 1996 Zambella [21] introduced an elegant way of presenting a hierarchy of second-order theories equivalent to $\langle S_2^i \rangle$, as well as the second-order theory P-def, which includes function symbols for all polynomial-time functions and is equivalent to QPV.

1.2 Descriptive Complexity

While in bounded arithmetic we are interested in the proving power of theories, in descriptive complexity we are interested in expressive power of formulas. Instead of asking what class of theorems can be proven, we ask what properties (for example, graph properties) we can express using certain classes of formulas. This research goes back to Fagin's 1974 result [7] showing that a language is in NP iff it corresponds to the set of finite models of an existential second-order formula. Later Stockmeyer [18] extended this result, characterizing the polynomial hierarchy as the class of sets of finite models of all second-order formulas.

Finding an elegant descriptive-style characterization of P proved more illusive. One such characterization of P uses the first-order logic augmented with the successor relation and the least fixed-point operator [20, 10]. Later Leivant [14, 15] found a second-order characterization of P using the notion of "controlled computational formula", which is related to Horn formula. (The motivation for using Horn formulas comes from the existence of a simple polynomial-time algorithm for solving the satisfiability problem for propositional Horn formulas.) Finally Grädel [8, 9] found an elegant descriptive characterization of P using $SO\exists$-Horn (second-order existential Horn) formulas with successor.

1.3 Our Results

We present a second-order theory V_1-Horn of bounded arithmetic based on Grädel's theorem; our theory is intended to capture polynomial-time reasoning. We use the elegant syntax of Zambella's [21] second-order theories. Our main new feature is a comprehension axiom scheme for second-order existential Horn formulas, which by Grädel's theorem represent the polynomial-time predicates. Our main results are that V_1-Horn is finitely axiomatizable, from which it follows that the $\forall\Sigma_1^b$ consequences of S_2^1 are finitely axiomatizable, thus answering an open question (see [12], theorem 10.1.2). The major tool needed for the proof is the construction in our theory of a Σ_1^B-Horn formula $\textsc{Run}_\Phi(R, \tilde{R})$ which encodes a run of the satisfiability algorithm on the Σ_1^B-Horn formula Φ (see Section 5).

Section 2 contains the background information. We define our system V_1-Horn in Section 3. In Section 4, we state that the syntactic definition of Σ_1^B-Horn formula can be considerably broadened with V_1-Horn proving the equivalence of each of the new formulas with a Σ_1^B-Horn formula, and demonstrate some techniques used for the proof. Section 5 contains the description of our main technique, namely, representing the Horn satisfiability algorithm in V_1-Horn by a Σ_1^B-Horn formula. In Section 6 we use this technique to show the equivalence of V_1-Horn and Zambella's P-def[21]. Finally, in Section 7 we demonstrate that our system is finitely axiomatizable, and show that this implies that $\forall\Sigma_1^b$ consequences of S_2^1 are finitely axiomatizable as well.

2 Second-order formulas and complexity classes

The prototype for the underlying language of V_1-Horn is the language of second-order bounded arithmetic introduced by Buss [1]. However our language is closer to the simpler second-order language introduced by Zambella [21].

Our language $\mathcal{L}_A^2$ has two sorts, first-order (i.e., natural numbers), denoted by lower-case letters and second-order (finite sets of natural numbers), denoted by upper-case letters. The first-order function and predicate symbols of $\mathcal{L}_A^2$ are the standard symbols $\{0, 1, +, \cdot; \leq, =\}$ of Peano Arithmetic, as well as the unary upper-bound function symbol $|\ |$, which takes second-order objects to first-order objects, and the binary membership predicate symbol $\in$.

We now define the classes Σ_i^B and Π_i^B of bounded second-order formulas to consist of formulas with i alternations of bounded second-order quantifiers starting with existential (universal), followed by any bounded first-order quantifiers. The classes Σ_i^B and Π_i^B are the formulas in our (Zambella's) simplified language $\mathcal{L}_A^2$ which correspond to the classes $\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ in Buss's prototype second-order language [1, 12]. They are the second-order analogies of the first-order formula classes Σ_i^b and Π_i^b, where sharply-bounded quantifiers correspond to our bounded first-order quantifiers.

The formulas Σ_1^B represent precisely the $\mathtt{NP}$ relations, and more generally for $i > 1$ the Σ_i^B formulas represent the Σ_i^p relations in the polynomial hierarchy and Π_i^B represent the Π_i^p relations [1, 12]. The formulas Σ_0^B represent precisely the uniform $\mathtt{AC}^0$ (that is, $\mathtt{FO}$ [11]) relations.

We now define the formulas corresponding to polynomial time.

Definition 2.1. A formula ϕ of $\mathcal{L}_A^2$ is *Horn with respect to the second-order variables* $P_1, ..., P_k$ if ϕ is quantifier-free in conjunctive normal form and in every clause there is at most one positive literal of the form $P_i(t)$ and no terms of the form $|P_i|$. (We *do* allow upper-bound terms $|X|$ and any number of positive literals $X(t)$, where X is not among $\{P_1, ..., P_k\}$.) A formula is Σ_1^B-Horn if it has the form

$$\exists P_1 ... \exists P_k \forall x_1 \leq t_1 ... \forall x_m \leq t_m \phi \tag{1}$$

where $k, m \geq 0$ and ϕ is Horn with respect to $P_1, ..., P_k$, and the bounding terms t_i do not involve $x_1, ..., x_m$. More generally a formula is *SO-Horn* if it has the above form except that each second-order quantifier can be either $\exists$ or $\forall$. A formula is Π_1^b *Horn with respect to* $P_1, ..., P_k$ if it has the form (1) with the existential quantifiers omitted.

Note that the second-order quantifiers $\exists P_i$ are not explicitly bounded in (1). However since $|P_i|$ may not occur in ϕ, they can be implicitly bounded by the maximum term u such that $P(u)$ occurs.

The second-order variables can be treated as multidimensional arrays using a standard pairing function to encode k-tuples of elements as one element.

The theorem below is similar to part of Grädel's Theorem 5.2 [8] (see also Chapter 7 of [17]), which is stated in the context of descriptive complexity theory.

Theorem 2.2. *A relation $R(z_1, ..., z_k, Y_1, ..., Y_m)$ is in $\mathtt{P}$ iff it is representable by a Σ_1^B-Horn formula Ψ. Further Ψ can be chosen with only one existentially quantified second-order variable, and only two universally quantified first-order variables.*

Example. (Parity(X)) This is a Σ_1^B-Horn formula which is true for strings X that contain an odd number of 1's. It encodes a dynamic-programming algorithm for computing parity of X: $P_{odd}(i)$ is true (and $P_{even}(i)$ is false) iff the prefix of X of length i contains an odd number of 1's.

$$\begin{aligned}
&\exists P_{even} \exists P_{odd} \forall i < |X| \\
&\qquad P_{even}(0) \wedge \neg P_{odd}(0) \wedge P_{odd}(|X|) \\
&\qquad \wedge(\neg P_{even}(i+1) \vee \neg P_{odd}(i+1)) \\
&\qquad \wedge(P_{even}(i) \wedge X(i) \rightarrow P_{odd}(i+1)) \wedge (P_{odd}(i) \wedge X(i) \rightarrow P_{even}(i+1)) \\
&\qquad \wedge(P_{even}(i) \wedge \neg X(i) \rightarrow P_{even}(i+1)) \wedge (P_{odd}(i) \wedge \neg X(i) \rightarrow P_{odd}(i+1))
\end{aligned}$$

Proof of theorem. The if direction uses Grädel's $SO\exists$-Horn satisfiability algorithm. First, an algorithm creates a propositional formula $PROP^{\Phi}$ by taking conjunction of quantifier-free part of the formula for all possible settings of first-order variables. Then, treating the second-order atoms as independent propositional variables, it executes Horn satisfiability algorithm.

The other direction is similar to the proof of Cook's theorem. □

Example. (3Color(n, E)) This is a Σ_1^B formula asserting that the graph with edge relation E on nodes $\{0, 1, ..., n-1\}$ is three-colorable. We write $E(x, y)$ like a binary relation, although it can be coded as a unary relation using the pairing function as explained above. The three colors are P, Q, and R.

$$\begin{aligned}
&\exists P \exists Q \exists R \forall x < n \forall y < n(P(x) \vee Q(x) \vee R(x)) \\
&\qquad \wedge(\neg E(x,y) \vee \neg P(x) \vee \neg P(y)) \wedge (\neg E(x,y) \vee \neg Q(x) \vee \neg Q(y)) \wedge (\neg E(x,y) \vee \neg R(x) \vee \neg R(y))
\end{aligned}$$

This formula is Σ_1^B-Horn except for the first clause. Since graph 3-colorability is NP-compete, it cannot be represented by a Σ_1^B-Horn formula unless P= NP. This example illustrates why we cannot allow bounded first-order existential quantifiers after the universal quantifiers in Σ_1^B-Horn formulas, since the first clause could be replaced by $\exists i < 3P(i, x)$ where now $P(0, x), P(1, x), P(2, x)$ represent the three colors.

3 V_1-Horn and other second-order theories

Our second-order theories use the language $\mathcal{L}_A^2$ described in the previous section. They all share the set 2-$BASIC$ of axioms in Table 1, which are similar to the axioms for Zambella's theory Θ [21] and form the second-order analog of Buss's first-order axioms $BASIC$ [1]. The set 2-$BASIC$ consists essentially of the axioms for Robinson's system Q, together with axioms for $\leq$, and two axioms defining the upper-bound terms $|X|$.

In addition to 2-$BASIC$, each system needs a comprehension scheme for some set $FORM$ of formulas.

$$FORM - \mathrm{COMP} : \exists X \leq y \forall z < y(X(z) \leftrightarrow \Phi(z)) \tag{2}$$

Here, Φ is any formula in the set $FORM$ with no free occurrence of X, but may have other free variables.

We denote by V^i the theory axiomatized by 2-$BASIC$ and Σ_i^B-COMP. For $i \geq 0$ V^i is essentially the same as Zambella's $\Sigma_i^p - comp$ [21]. For $i \geq 1$, V^i is like V_1^i [12]. Thus for $i \geq 1$ V^i is a second-order version of S_2^i. In particular, the Σ_1^B-definable functions in V^1 are precisely the polynomial-time functions [4]. The Σ_1^B-definable functions in V^0 are the AC^0 functions [4] (called *rudimentary functions* in [21]).

Robinson's Theory Q axioms	B1	$x+1 \neq 0$	B2	$x+1=y+1 \rightarrow x=y$
	B3	$x+0=x$	B4	$x+(y+1)=(x+y)+1$
	B5	$x \cdot 0=0$	B6	$x \cdot (y+1)=(x \cdot y)+x$
Axioms for $\leq$	B7	$0 \leq x$	B8	$x \leq x+y$
	B9	$x \leq y \wedge y \leq z \rightarrow x \leq z$	B10	$(x \leq y \wedge y \leq x) \rightarrow x=y$
	B11	$x \leq y \vee y \leq x$	B12	$x \leq y \leftrightarrow x < y+1$
Predecessor	B13	$x \neq 0 \rightarrow \exists y(y+1=x)$		
Upper bound	L1	$X(y) \rightarrow y < \|X\|$	L2	$y+1=\|X\| \rightarrow X(y)$

Table 1: The 2-$BASIC$ Axioms

Definition 3.1. V_1-Horn is the theory axiomatized by 2-$BASIC$ and Σ_1^B-Horn-COMP.

Although 2-$BASIC$ does not include an explicit induction axiom, L2 asserts that a nonempty set has a largest element. This can be turned into a least number principle, from which induction follows. (Thus V_1-Horn proves the induction scheme for Σ_1^B-Horn formulas.) Other simple properties, including k-ary comprehension and replacement, are also easy to prove.

4 Formulas provably equivalent to Σ_1^B-Horn

We can now proceed to show that our class is robust, in a sense that some other classes of formulas are representable as Σ_1^B-Horn. In particular, we can show that for Σ_0^B formulas, as well as Σ_1^B-Horn formulas preceded by several alternating second-order quantifiers, there are (provably in V_1-Horn) equivalent Σ_1^B-Horn formulas.

A major inconvenience of Σ_1^B-Horn formulas is lack of first-order existential quantifiers. In general we cannot allow such quantifiers without increasing the apparent expressive power of the formulas, as pointed out in the 3-colorability example. However, it is possible to introduce bounded existential quantifiers in some contexts.

Notation. If P is a second-order variable, then $\tilde{P}$ denotes a second-order variable whose intended interpretation is $\neg P$. □

We now introduce the Horn formulas $\textsc{Search}_k$, which are Π_1^b Horn with respect to all of their second-order variables and which will allow a SO∃-Horn formula to represent $\exists z < bX(\bar{y}, z)$. $\textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$ asserts that $S(\bar{y}, i)$ holds iff $X(\bar{y}, z)$ holds for some $z < i$, where $\bar{b}$ stands for $b_1, ..., b_k$, and $\bar{y}$ stands for $y_1, ..., y_k$. We use $\bar{y} < \bar{b}$ for $y_1 < b_1 \wedge ... \wedge y_k < b_k$.

Definition 4.1. For each $k \geq 1$ $\textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$ is the Π_1^b Horn formula

$$\begin{aligned} \forall \bar{y} < \bar{b} \forall i < b(&\neg S(\bar{y}, 0) \wedge \tilde{S}(\bar{y}, 0)) \\ &\wedge(\neg S(\bar{y}, i+1) \vee \neg \tilde{S}(\bar{y}, i+1)) \\ &\wedge(S(\bar{y}, i) \rightarrow S(\bar{y}, i+1)) \\ &\wedge(X(\bar{y}, i) \rightarrow S(\bar{y}, i+1)) \\ &\wedge(\tilde{S}(\bar{y}, i) \wedge \tilde{X}(\bar{y}, i) \rightarrow \tilde{S}(\bar{y}, i+1)) \end{aligned}$$

Lemma 4.2. *V_1-Horn proves the following:*

$$(i)\ \ \forall z < b(X(\bar{y},z) \leftrightarrow \neg\tilde{X}(\bar{y},z)) \wedge \bar{y} < \bar{b} \ \rightarrow\ \exists S \exists \tilde{S}\ \textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X})$$

$$\begin{aligned}(ii)\ &\forall z < b(X(\bar{y},z) \leftrightarrow \neg\tilde{X}(\bar{y},z)) \wedge \textsc{Search}_k(\bar{b}, b, S, \tilde{S}, X, \tilde{X}) \wedge \bar{y} < \bar{b}\\ &\rightarrow (S(\bar{y},b) \leftrightarrow \exists z < bX(\bar{y},z)) \wedge (\tilde{S}(\bar{y},b) \leftrightarrow \forall z < b\tilde{X}(\bar{y},z))\end{aligned}$$

Proof. Omitted. □

5 Encoding the Horn SAT algorithm by a Σ_1^B-Horn formula

Here we show that a run of the polynomial-time algorithm described in the proof of Theorem 2.2 can be represented by a Σ_1^B-Horn formula $\textsc{Run}$. This result is needed for sections 6 and 7. A simple corollary is that the negation of a Σ_1^B-Horn formula is provably equivalent to a Σ_1^B-Horn formula. In other words, V_1-Horn proves that P is closed under complementation.

Theorem 5.1. *Let Φ be a Σ_1^B-Horn formula which does not involve R or $\tilde{R}$. Then there is a formula $\textsc{Run}_\Phi(R,\tilde{R})$ whose free variables include those of Φ in which the only atomic subformulas involving R and $\tilde{R}$ are $R(0)$ and $\tilde{R}(0)$ and such that $\exists R \exists \tilde{R} \textsc{Run}_\Phi(R,\tilde{R})$ is a Σ_1^B-Horn formula and V_1-Horn proves the following:*

$$(i)\ \ \exists R \exists \tilde{R}\ \textsc{Run}_\Phi(R,\tilde{R})$$
$$(ii)\ \textsc{Run}_\Phi(R,\tilde{R}) \rightarrow [(R(0) \leftrightarrow \Phi) \wedge (\tilde{R}(0) \leftrightarrow \neg\Phi)]$$

Corollary 5.2. *If Φ is Σ_1^B-Horn, then $\neg\Phi$ is provably equivalent in V_1-Horn to a Σ_1^B-Horn formula $\textsc{Neg}_\Phi$.*

Corollary 5.3. *The class of formulas provably equivalent in V_1-Horn to a Σ_1^B-Horn formula is closed under $\neg$, $\wedge$, $\vee$, and bounded first-order quantification.*

Theorem 5.1 can be generalized to the case in which arrays $R(\bar{y})$ and $\tilde{R}(\bar{y})$ code values of $\Phi(\bar{y})$ and $\neg\Phi(\bar{y})$.

Corollary 5.4. *Let $\Phi(\bar{y})$ be a Σ_1^B-Horn formula which does not involve R or $\tilde{R}$. Then there is a formula $\textsc{Run}_{\Phi(\bar{y})}(\bar{b}, R, \tilde{R})$ which does not have $\bar{y}$ free but whose free variables include any other free variables of Φ such that $\exists R \exists \tilde{R} \textsc{Run}_{\Phi(\bar{y})}(R,\tilde{R})$ is a Σ_1^B-Horn formula and V_1-Horn proves the following:*

$$(i)\ \ \exists R \exists \tilde{R}\ \textsc{Run}_{\Phi(\bar{y})}(\bar{b}, R, \tilde{R})$$
$$(ii)\ \textsc{Run}_{\Phi(\bar{y})}(\bar{b}, R, \tilde{R}) \rightarrow \forall \bar{y} < \bar{b}[(R(\bar{y}) \leftrightarrow \Phi(\bar{y})) \wedge (\tilde{R}(\bar{y}) \leftrightarrow \neg\Phi(\bar{y}))]$$

Note that it suffices to prove Theorem 5.1 for Σ_1^B-Horn formulas of the form

$$\Phi \equiv \exists P \forall x_1 \leq t_1 ... \forall x_k \leq t_k \phi(\bar{x}, P) \tag{3}$$

where ϕ is Horn with respect to P.

The algorithm we wish to represent has two main steps (see the proof of Theorem 2.2): First create a propositional Horn formula $\textsc{Prop}^\Phi$, and second apply the Horn Sat algorithm to determine whether $\textsc{Prop}^\Phi$ is satisfiable. We represent $\textsc{Prop}^\Phi$ using the arrays C, D, V, and we will present a

Σ_1^B-Horn formula $\text{PROP}_\Phi(C, \tilde{C}, D, \tilde{D}, V, \tilde{V})$ which defines these arrays and their negations. Besides the indicated free variables, PROP_Φ also has as free variables the free variables of Φ. For the second step we present a Σ_1^B-Horn formula $\text{HORNSAT}(a, b, C, \tilde{C}, D, \tilde{D}, V, \tilde{V}, R, \tilde{R})$ (with all free variables indicated) which is independent of Φ and which sets the result variable $R(0)$ true iff PROP^Φ is satisfiable.

The arrays C, D, V together with the scalars a, b completely specify the formula PROP^Φ as follows. The atoms of PROP^Φ are $P(0), ..., P(a-1)$, and the clauses are $cl_0, ..., cl_{b-1}$. We allow both the empty clause and the special clause TRUE. The arrays C, D, V are defined as follows: For $0 \leq x < b$, $0 \leq v < a$, they assert that clause cl_x

- $C(x, v)$: contains the negative literal $\neg P(v)$.
- $D(x, v)$: contains the positive literal $P(v)$.
- $V(x)$: is the clause TRUE.

These assertions can be represented by Σ_0^B formulas. Since PROP^Φ is a Horn formula, for each x, $D(x, v)$ can be true for at most one v.

Lemma 5.5. *$\text{PROP}_\Phi(\bar{Q})$ can be defined in such a way that $\exists\bar{Q}\text{PROP}_\Phi(\bar{Q})$ is Σ_1^B-Horn and V_1-Horn proves*

$$(i)\ \exists\bar{Q}\text{PROP}_\Phi(\bar{Q})$$

$$(ii)\ \text{PROP}_\Phi(\bar{Q}) \ \rightarrow\ \forall v < \hat{a}\forall x < \hat{b}$$
$$[(C(x,v) \leftrightarrow \psi_C(x,v)) \wedge (D(x,v) \leftrightarrow \psi_D(x,v)) \wedge (V(x) \leftrightarrow \psi_V(x))$$
$$\wedge\ (\tilde{C}(x,v) \leftrightarrow \neg\psi_C(x,v)) \wedge (\tilde{D}(x,v) \leftrightarrow \neg\psi_D(x,v)) \wedge (\tilde{V}(x) \leftrightarrow \neg\psi_V(x))]$$

The array bounds a, b are represented by terms $\hat{a}, \hat{b}$ in the free variables of Φ and are determined as follows. For each term s in $\phi(\bar{x}, P)$ in (3) let $\hat{s}$ be the result of replacing each variable $x_1, ..., x_k$ by its respective upper bound $t_1, ..., t_k$. Then the upper bound $\hat{a}$ on the arguments of $P()$ is

$$\hat{a} \equiv \hat{s}_1 + ... + \hat{s}_\ell$$

where $s_1, ..., s_\ell$ is a list of all terms such that $P(s_i)$ or $\neg P(s_i)$ occurs in Φ.

The upper bound $\hat{b}$ on the number of clauses in PROP^Φ is

$$\hat{b} \equiv \langle t_1, ..., t_s, m\rangle$$

where $t_1, ..., t_s$ are as in (3), m in the number of clauses in $\phi(\bar{x}, P)$, and $\langle ... \rangle$ is the tupling function.

Using the abbreviation

$$\bar{Q} \equiv C, \tilde{C}, D, \tilde{D}, V, \tilde{V}$$

we can now choose $\text{RUN}_\Phi(R, \tilde{R})$ to be a Σ_1^B-Horn formula such that

$$\text{RUN}_\Phi(R, \tilde{R}) \ \leftrightarrow\ \exists\bar{Q}[\text{PROP}_\Phi(\bar{Q}) \wedge \text{HORNSAT}(\hat{a}, \hat{b}, \bar{Q}, R, \tilde{R})] \tag{4}$$

In fact we take $\text{RUN}_\Phi(R, \tilde{R})$ to be a suitable prenex form of the right hand side.

5.1 Definition of HornSat$(a, b, C, \tilde{C}, D, \tilde{D}, V, \tilde{V}, R, \tilde{R})$

The algorithm represented by $\textsc{HornSat}(a, b, \bar{Q}, R, \tilde{R})$ attempts to find a satisfying assignment to the Horn formula $\textsc{Prop}^{\Phi}$ described by the parameters a, b, C, D, V. This is done by filling in an array $T(t, v)$, where $T(t, v)$ is the truth value assigned to the atom $P(v)$ after step t, $0 \leq t, v < a$. Initially $T(0, v)$ is false, and at step $t+1$ $T(t+1, v)$ sets each $P(v)$ true such that $P(v)$ occurs positively in some clause not satisfied after step t. Once $P(v)$ is set true, it is never changed to false.

That is, if $\textsc{Init} \equiv \forall v < a(\tilde{T}(0, v) \wedge \neg T(0, v))$ is the initialization of T, and $Step(v, T_1)$ encodes one step of the algorithm, then the run of the HornSat algorithm is encoded by

$$\textsc{TDef}(a, b, \bar{Q}, T, \tilde{T}) \equiv \textsc{Init}(T, \tilde{T}) \wedge \forall t < a \exists \bar{W} \textsc{Step}^*(a, T^{[t+1]}, \tilde{T}^{[t+1]}, \bar{W}, T^{[t]}, \tilde{T}^{[t]})$$

Here, $T^[t]$ notation means t^{th} row of two-dimensional array T.

We define $\textsc{Sat}(T_1)$ to assert that the truth assignment T_1 satisfies $\textsc{Prop}^{\Phi}$. This can be used to check that the truth assignment obtained after the run of HornSat algorithm satisfies the original formula. We also need a formula $\textsc{Neg}$ asserting that $\tilde{C}, \tilde{D}, \tilde{V}$ and $\tilde{T}$ correspond to negations of C, D, V and T, respectively. Abbreviating C, D, V and their negations as Q, we can set

$$\textsc{HornSat}(a, b, \bar{Q}, R, \tilde{R}) \equiv \exists T \exists \tilde{T}[\textsc{TDef}(a, b, \bar{Q}, T, \tilde{T}) \wedge \exists \bar{W} \textsc{Sat}^*(1, R, \tilde{R}, \bar{W}, \bar{Q}, T^{[a]}, \tilde{T}^{[a]})] \quad (5)$$

Note that no part of the formula other than $\textsc{Prop}_{\Phi}$ depends on the original Φ; the rest depends only on the encodings C, D and V. This fact is important in the proof of the finite axiomatizability of V_1-Horn(Section 7).

6 Equivalence of V_1-Horn P-def and QPV

The first-order theory QPV (called PV1 in [12]) has function symbols for all polynomial-time computable functions, and the axioms include defining equations for these functions (based on Cobham's Theorem) and induction on the length of numbers. The theory has been extensively studied [3, 1, 6, 12, 5] and shown to robustly capture the notion of "polynomial-time reasoning". Zambella's [21] theory P-def is a second-order version of QPV, and can shown to be equivalent to QPV by the method of RSUV isomorphism (see [12]). Here we show that V_1-Horn is equivalent in power to P-def. This implies that V_1-Horn is equivalent in power to QPV, but is most likely not as powerful as S_2^1 (see Section 1). We begin by showing how to add function symbols to V_1-Horn.

6.1 Adding function symbols for polynomial-time functions to V_1-Horn

The class P can be defined in our second-order setting to consist of relations $R(x_1, ..., x_k, Y_1, ..., Y_m)$ recognizable in time bounded by a polynomial in $(x_1, ..., x_k, |Y_1|, ..., |Y_m|)$. Similarly, we now define the class FP to consist of all functions $F(x_1, ..., x_k, Y_1, ..., Y_m)$ computable in time bounded by a polynomial in $(x_1, ..., x_k, |Y_1|, ..., |Y_m|)$. There are two kinds of functions in FP; *string functions*, denoted by upper-case letters F, take second-order objects as values, and *number* functions, denoted by lower-case letters f, take first-order objects as values. As before, number values are expressed in unary notation when defining computation time.

We say that a function has *arity* $\langle k, m \rangle$ if it takes k number arguments and m string arguments.

It is convenient to represent a string function $F(\bar{x}, \bar{Y})$ by its *bit graph* $B_F(\bar{x}, \bar{Y})$, defined by the condition

$$B_F(i, \bar{x}, \bar{Y}) \Leftrightarrow F(\bar{x}, \bar{Y})(i)$$

That is, $B_F(i, \bar{x}, \bar{Y})$ holds iff the i-th bit of $F(\bar{x}, \bar{Y})$ is 1. The following characterization of FP is straightforward.

Lemma 6.1. *(i) A string function $F(\bar{x}, \bar{Y})$ is in* FP *iff $|F(\bar{x}, \bar{Y})|$ is bounded by a polynomial in $(\bar{x}, |\bar{Y}|)$ and its bit graph B_F is in* P.
(ii) A number function $f(\bar{x}, \bar{Y})$ is in FP *iff $f(\bar{x}, \bar{Y}) = |F(\bar{x}, \bar{Y})|$ for some string function F in* FP.

We now define a conservative extension V_1-Horn(FP) of V_1-Horn by introducing function symbols for polynomial time functions with defining equations based on the above Lemma. These function symbols are ordered, and the defining equations for later symbols may involve earlier symbols. Definition 2.1 of a Σ_1^B-Horn formula Φ is generalized to allow these new function symbols, provided that no term of Φ includes a quantified second-order variable P_i as a proper subpart. (This generalizes the restriction that $|P_i|$ may not appear.) However formulas $P_i(t)$ may appear for any term t satisfying this restriction. With this restriction, the proof of Theorem 2.2 is easily modified to show that these generalized Σ_1^B-Horn formulas still represent only polynomial time relations.

The axioms of V_1-Horn(FP) are the same as for V_1-Horn except that the comprehension scheme is generalized to allow comprehension for all Σ_1^B-Horn formulas of V_1-Horn(FP), and the defining formulas for all function symbols are included.

It is not hard to check that the results in the previous two sections apply to V_1-Horn(FP) as well as to V_1-Horn. This is true in particular to the main theorem on $\textsc{Run}_\Phi$.

The main result of this subsection states that function symbols can be removed from generalized Σ_1^B-Horn formulas.

Theorem 6.2. *For every Σ_1^B-Horn formula $\Phi'(\bar{x}, \bar{Y})$ of V_1-Horn(*FP*) there is a Σ_1^B-Horn formula Φ of V_1-Horn such that*

$$V_1\text{-}Horn(\mathtt{FP}) \vdash \Phi'(\bar{x}, \bar{Y}) \leftrightarrow \Phi(\bar{x}, \bar{Y})$$

The proof is a tricky induction on the complexity of the terms in Φ', and on the order of the function symbols.

Corollary 6.3 (Conservativity). *Every theorem of V_1-Horn(*FP*) in the language of V_1-Horn is a theorem of V_1-Horn.*

6.2 Specification of P-def

We present a version of Zambella's [21] P-def which fits our notation and axioms. It is the same in spirit to Zambella's system. The system P-def is obtained from a *Base Theory BT* by introducing function symbols for all functions in FP, based on Cobham's recursion-theoretic characterization of the polynomial-time computable functions.

The Base Theory BT has the language $\mathcal{L}_A^2(=)$, which is $\mathcal{L}_A^2$ with second-order equality. System BT has the same terms and formulas as V_1-Horn, except that atomic formulas include equations $X = Y$ between second-order variables. The axioms of BT consist of the axioms B1,...,B13,L1,L2 of V_1-Horn, the axiom E of extensionality (below) and the comprehension scheme for Σ_0^B formulas.

$$\text{E}: \qquad X = Y \leftrightarrow [\ |X| = |Y| \wedge \forall i < |X|(X(i) \leftrightarrow Y(i))] \tag{6}$$

To obtain P-def, first the (string and number) function symbols for all $\mathtt{AC}^0$ functions (i.e., with Σ_0^B formulas as bit graphs) are added to the language. Then, new function symbols are introduced using limited recursion on notation; that gives all functions in FP. Now the axioms of P-def are $B11, \ldots, B13, L1, L2$ and E, together with Σ_0^B comprehension over formulas of P-def, as well as defining formulas for all function symbols.

Now, using the translations of function symbols, we can prove that

Theorem 6.4. *Every formula of V_1-Horn is a theorem of V_1-Horn iff it is a theorem of* P*-def.*

7 Finite Axiomatizability of V_1-Horn and its $\forall\Sigma_1^B$ Fragment

Theorem 7.1. *V_1-Horn is axiomatized by its $\forall\Sigma_1^B$ consequences.*

Proof. It suffices to show that each Σ_1^B-Horn comprehension axiom is a consequence of $\forall\Sigma_1^B$ theorems of V_1-Horn. First note that for each Σ_1^B-Horn formula Φ there is a Σ_1^B formula Φ^B such that $\forall\Sigma_1^B V_1\text{-Horn} \vdash (\Phi \leftrightarrow \Phi^B)$.

The comprehension axiom (2) for $\Phi(z)$ follows from Corollary 5.4 (i) and (ii). The Σ_1^B form of (i) we need is

$$\exists R \leq y \exists \tilde{R} \leq y \ \textsc{Run}'_{\Phi(z)}(y, R, \tilde{R})$$

where $\textsc{Run}'_{\Phi(z)}$ has suitable bounds on its second-order quantifiers. For (ii) we do not need the clause involving $\tilde{R}$. If we replace Φ by Φ^B then a suitable prenex form of the result is $\forall\Sigma_1^B$. □

Lemma 7.2. *V^0 is finitely axiomatizable.*

Proof. We must show that all Σ_0^B-COMP axioms follow from finitely many theorems of V^0 (see section 3). We show that each $\Phi(z)$-COMP axiom $\exists X \leq y \forall z < y(X(z) \leftrightarrow \Phi(z))$ can be derived, by structural induction on the Σ_0^B formula $\Phi(z)$, introducing finitely many axioms along the way. □

Theorem 7.3. *V_1-Horn is finitely axiomatizable.*

Proof. It suffices to show that Corollary 5.4 (i) and (ii) can be proved for any Σ_1^B-Horn formula $\Phi(y)$ using finitely many theorems of V_1-Horn as axioms. We first will show how to do this for Theorem 5.1 (i) and (ii), and then explain how to modify the proof to get the corollary.

First note that for each Σ_1^B-Horn formula Φ we can define a version of $\textsc{Prop}_\Phi$ such that (i) and (ii) in Lemma 5.5 are theorems of V^0. The proof of Theorem 5.1 depends on Lemma 5.5 (which we have established) and some properties of HornSat. Since HornSat is independent of Φ, we can take these properties as axioms.

To generalize the proof of Theorem 5.1 in order to prove Corollary 5.4, we incorporate the variable y in $\Phi(y)$ as an argument of each of the arrays $C, D, V, \tilde{C}, \tilde{D}, \tilde{V}$ to define the formula $\textsc{Prop}_\Phi(y)$ in a modified Lemma 5.5. Then y is not free in $\textsc{Prop}_\Phi(y)$ (although it could be free in $\textsc{Prop}_\Phi$). The definition (5) of HornSat is modified so that the parameter y is incorporated as an argument of each of the arrays $R, \tilde{R}, T, \tilde{T}$. Then Corollary 5.4 follows in the same way as Theorem 5.1. □

Corollary 7.4. *The $\forall\Sigma_1^B$ consequences of V_1-Horn are finitely axiomatizable. The $\forall\Sigma_1^b$ consequences of S_2^1 are finitely axiomatizable.*

Proof. The first sentence follows by compactness from Theorems 7.1 and 7.3. Since V^1 is $\forall\Sigma_1^B$ conservative over P-def [21], it follows from Theorem 6.4 that the $\forall\Sigma_1^B$ consequences of V^1 and of V_1-Horn are the same, and hence are finitely axiomatizable. The second sentence of the Corollary is equivalent to asserting that the $\forall\Sigma_1^B$ consequences of V^1 are finitely axiomatizable, by the RSUV isomorphism. □

References

[1] S. Buss. *Bounded Arithmetic.* Bibliopolis, Naples, 1986.

[2] S. Buss. Relating the bounded arithmetic and polynomial time hierarchies. *Annals of Pure and Applied Logic*, 75:67–77, 1995.

[3] S. A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 83 –97, 1975.

[4] S. A. Cook. CSC 2429S: Proof Complexity and Bounded Arithmetic. Course notes, URL: "http://www.cs.toronto.edu/~sacook/csc2429h", Spring 1998.

[5] S. A. Cook. Relating the provable collapse of P to NC^1 and the power of logical theories. *DIMACS series in Discrete mathematics and theoretical computer science*, 39:73–91, 1998.

[6] S. A. Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103 – 200, 1993.

[7] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of computation, SIAM-AMC proceedings*, 7:43–73, 1974.

[8] E. Grädel. The Expressive Power of Second Order Horn Logic. In *Proceedings of 8th Symposium on Theoretical Aspects of Computer Science STACS '91, Hamburg 1991*, volume 480 of *LNCS*, pages 466–477. Springer-Verlag, 1991.

[9] E. Grädel. Capturing Complexity Classes by Fragments of Second Order Logic. *Theoretical Computer Science*, 101:35–57, 1992.

[10] N. Immerman. Relational queries computable in polytime. *Information and Control*, 68:86 –104, 1986.

[11] N. Immerman. *Descriptive complexity.* Springer Verlag, New York, 1999.

[12] J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, New York, USA, 1995.

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

[14] Daniel Leivant. Characterization of complexity classes in higher-order logic. In *Proceedings of the Second Annual Conference on Structure in Complexity Theory*, pages 203–217, 1987.

[15] Daniel Leivant. Descriptive characterizations of computational complexity. *Journal of Computer and System Sciences*, 39:51–83, 1989.

[16] A. Razborov. An equivalence between second-order bounded domain bounded arithmetic and first-order bounded arithmetic. In P. Clote and J. Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 247–277. Clarendon Press, Oxford, 1993.

[17] U. Schöning and R. Pruim. *Gems of theoretical computer science.* Springer, Berlin, 1998.

[18] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[19] G. Takeuti. RSUV isomorphism. In P. Clote and J. Krajíček, editors, *Arithmetic, proof theory and computational complexity*, pages 364–386. Clarendon Press, Oxford, 1993.

[20] M. Vardi. Complexity of relational query languages. *Information and Control*, 68:137 –146, 1986.

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