DIMACS Series in Discrete Mathematics
and Theoretical Computer Science
Volume 00, 19xx

Relating the Provable Collapse of P to NC^1 and the Power of Logical Theories

Stephen Cook

ABSTRACT. We show that the following three statements are equivalent: **QPV** is conservative over **QALV**, **QALV** proves its open induction formulas, and **QALV** proves $\mathbf{P} = \mathbf{NC}^1$. Here **QPV** and **QALV** are first order theories whose function symbols range over polynomial-time and $\mathbf{NC}^1$ functions, respectively.

1. Introduction

The motivation for this paper comes from two sources. First, building on a result of Krajíček , Pudlák , and Takeuti [**KPT91**], Buss [**Bus95**] and Zambella [**Zam96**] independently proved that the bounded arithmetic hierarchy of theories $\mathbf{S}_2^i$ collapses if and only if the polynomial time hierarchy provably collapses. We prove that the arithmetic theory **QPV** of polynomial time functions is conservative over the arithmetic theory **QALV** of $\mathbf{NC}^1$ functions if and only if **P** provably collapses to $\mathbf{NC}^1$. (**QPV** is called $\mathbf{PV}_1$ in [**Bus95**].) The second motivation is an attempt to relate the possible collapse of **P** to $\mathbf{NC}^1$ with the possible p-simulation of extended Frege propositional proof systems by Frege systems.

The theory **QPV** considered here is a quantified version of the author's equational theory **PV** [**Coo75**], in which the function symbols represent the polynomial time functions. **PV** is well-studied [**CU93, Kra95a**], and in particular Buss [**Bus86**] showed that the theorems of **PV** are inter-translatable with the Σ_1^b theorems of $\mathbf{S}_2^1$.

The theory **QALV** is some analog of **QPV** for the $\mathbf{NC}^1$ functions; i.e. the functions computable by uniform log depth polynomial size circuit families. By a theorem of Ruzzo [**Ruz81**], the $\mathbf{NC}^1$ functions are the same as the **ALOGTIME** functions; the functions whose i-th bit relations are computable by alternating Turing machines in log time. There are several possible choices for **QALV**. The one we select is a quantified version of Clote's equational theory $\mathbf{ALV}'$ [**Clo93**]. Other possibilities could be based on Pitt's quantifier-free theory $\mathbf{T}_1$ [**Pit96**], or Arai's first-order theory **AID** [**Ara91**]. It may turn out that all these theories are essentially equivalent, but so far this is an unproven conjecture.

1991 *Mathematics Subject Classification.* Primary 03F30; Secondary 68Q15, 03F20.

Research supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Information Technology Research Centre, Ontario (ITRC).

The main reason for selecting $\mathbf{ALV}'$ is that it closely resembles $\mathbf{PV}$, and in fact $\mathbf{PV}$ can be regarded naturally as an extension of $\mathbf{ALV}'$. ¿From our point of view, the disadvantage of $\mathbf{T}_1$ is that it is a theory of strings rather than numbers, and the disadvantage of $\mathbf{AID}$ is that, although its predicate symbols range over all $\mathbf{NC}^1$ relations, it has only finitely many function symbols.

A possible disadvantage of $\mathbf{ALV}'$ is that the proof that its function symbols represent all $\mathbf{NC}^1$ functions rests on Barrington's theorem [**Bar89**], which involves an interesting construction using unsolvable groups. Thus functions as simple as the majority function require relatively complex definitions, and proofs of their properties appear to be complex. However it turns out that the functions we need in this paper are relatively simple and can be represented in $\mathbf{ALV}'$ without Barrington's construction. Nevertheless it remains an open question whether $\mathbf{QALV}$ (quantified $\mathbf{ALV}'$) is equivalent, for example, to $\mathbf{AID}$.

A fundamental open question in propositional proof complexity is whether Frege systems can efficiently simulate extended Frege systems [**CR79, Kra95a**]. Intuitively this question is related to the question of whether $\mathbf{P} = \mathbf{NC}^1$, since formulas in a Frege proof represent $\mathbf{NC}^1$ concepts and formulas in an extended Frege proof represent polynomial time concepts. However, no formal relation between the two questions has been proven.

There are known relationships between the power of logical theories and the power of Frege and extended Frege proof systems. In general, $\forall$-theorems of theories which capture reasoning with polynomial time (respectively, $\mathbf{NC}^1$) concepts correspond to sequences of tautologies with polynomial size extended Frege (respectively, Frege) proofs, and such theories prove the soundness of extended Frege (respectively, Frege) systems.

In the case of polynomial time concepts, the author [**Coo75**] carried out this correspondence for his equational theory $\mathbf{PV}$, and argued that $\mathbf{PV}$ proves the soundness of extended Frege systems. The correspondence has been extended to show that the $\forall\Sigma_1^b$-theorems of Buss's first order theory $\mathbf{S}_2^1$ and the $\forall\Sigma_0^{1,b}$ theorems of the second order theory $\mathbf{V}_1^1$ correspond to tautology sequences with polynomial size extended Frege proofs [**Kra95b, Kra95a**].

In the case of $\mathbf{NC}^1$ concepts, several authors have results. Clote [**Clo90, Clo93**] showed that the theorems of his equational theories $\mathbf{ALV}$ and $\mathbf{ALV}'$ correspond to tautology sequences with polynomial size Frege proofs. Arai [**Ara91**] did the same for the Σ_0^b theorems of his first order theory $\mathbf{AID}$ and Pitt [**Pit96**] did the same for his quantifier-free theory $\mathbf{T}_1$. Krajicek [**Kra95b**] showed that every $\forall\Sigma_0^{1,b}$ theorem of the second order theory $\mathbf{U}_1^1$ corresponds to a sequence of tautologies with quasi-polynomial size Frege proofs, but apparently no one has yet defined a satisfactory second order theory that nicely corresponds to polynomial size Frege proofs (see [**Kra95b**], page 297).

Buss [**Bus91**] proved that Frege systems have polynomial size partial self-consistency proofs. Arai showed how to formalize this in his system $\mathbf{AID}$, to show that $\mathbf{AID}$ proves the soundness of Frege proof systems. Buss also showed that Frege systems have polynomial size proofs of partial consistency of extended Frege systems iff Frege systems p-simulate extended Frege systems. It seems plausible that this result can also be formalized in $\mathbf{AID}$. Assuming this and a suitable interpretation of $\mathbf{AID}$ in $\mathbf{PV}$, we would have

Assertion: The following statements are equivalent:

1. **QPV** is $\forall\Pi_1^b$-conservative over **AID**.
2. **AID** proves the soundness of extended Frege systems.
3. **AID** proves that Frege systems p-simulate extended Frege systems.

Proof sketch: $1 \Longrightarrow 2$: **QPV** proves the soundness of extended Frege systems [**Coo75**], and using the methods of [**Bus91**] and [**Ara91**] this soundness assertion can be formulated as a $\forall\Pi_1^b$-sentence in **AID**.

$2 \Longrightarrow 3$: Formalize Buss's theorem [**Bus91**] in **AID**.

$3 \Longrightarrow 1$: Let A be a $\forall\Pi_1^b$-formula of **AID** which is provable in **QPV**. Then by results in [**Coo75**] (see [**Kra95a**]), A gives rise to a sequence of propositional tautologies which have polynomial size extended Frege proofs. Even though the **QPV** proof involves polynomial time concepts which may not be in **NC**^{1}, these extended Frege proofs are uniform and can be described by **NC**^{1} functions, and there existence can be proved in **AID**. By assumption 3, **AID** proves that these tautologies have polynomial size Frege proofs. Since **AID** proves the soundness of Frege systems, **AID** proves A. □

It would be very interesting to show that a fourth statement is equivalent to the above three: **AID** proves that **P** = **NC**^{1}. However by Theorem 7 below this would be the same (assuming that **AID** and **QALV** are equivalent theories: see above) as showing that Π_1^b-conservativity of **QPV** over **AID** is equivalent to Σ_1^b-conservativity. Although in general the Σ_1^b-conservativity of theories is not the same as Π_1^b conservativity, it seems a plausible conjecture in this case.

The hard part of the proof of our main theorem (Theorem 7) is showing that if **QALV** proves its open induction formulas, then **P** = **NC**^{1}, and this equality is provable in **QALV**. At a high level, our proof resembles Buss's [**Bus95**] proof that if $\mathbf{S}_2^1(\mathbf{PV}) = \mathbf{QPV}$, then the polynomial hierarchy collapses. Buss's proof is based on the KPT witnessing theorem [**KPT91**], applied to a $\forall\exists\forall$ formula. Our proof is based on a simpler traditional witnessing theorem, applied to a $\forall\exists$ formula (induction).

2. The theories QPV and QALV

2.1. The equational theories. We start by presenting an outline of the equational theory **PV** following [**CU93**]. The function symbols of **PV** stand for polynomial time computable functions (members of **FP**), and every such function is represented by infinitely many function symbols. The function symbols are introduced in a manner based on Cobham's [**Cob65**] characterization of **FP** as the least class including certain initial functions and closed under composition and bounded recursion on notation.

In more detail, the *initial* function symbols of **PV** are 0, s_0, s_1, $parity$, tr, $cond$, pad, $trunc$, and $\#$. The intended interpretations of these are $s_0(x) = 2x$, $s_1(x) = 2x+1$, $parity(x) = x \bmod 2$, $tr(x) = \lfloor x/2 \rfloor$, $cond(0,y,z) = y$ and $cond(x,y,z) = z$ for $x > 0$, $pad(x,y) = 2^{|y|} \cdot x$, where $|y| = \lceil \log_2(y+1) \rceil$, $trunc(x,y) = \lfloor x/2^{|y|} \rfloor$, and $x \# y = 2^{|x| \cdot |y|}$.

In general, if t is a **PV** term, and $x_1, ..., x_n$, $n \geq 0$, is a list of variables including all variables in t, then $[\lambda x_1 ... x_n . t]$ is an n-place function symbol. If g, h, and k are n-place, $n+2$-place, and $n+1$-place function symbols respectively, then $R[g,h,k]$

is an $n+1$-place function symbol, representing the result of bounded recursion on notation from g, h, and k.

The axioms of **PV** consist of $s_0(0) = 0$, together with one, two, or three *defining equations* for each function symbol, except s_0 and s_1 have no defining equation. The defining equations for the remaining initial functions characterize their meaning via recursion on notation. The rules of **PV** are the usual rules for equality, together with induction on notation.

See Appendix A for a complete list of axioms and rules.

Now we present an outline of the equational theory $\mathbf{ALV}'$, following [**Clo93**]. $\mathbf{ALV}'$ is similar to **PV**, except that the function symbols range over $\mathbf{NC}^1$ functions instead of polynomial time functions. The $\mathbf{NC}^1$ function symbols are introduced in a manner based on Barrington's characterization of $\mathbf{NC}^1$ in terms of bounded width branching programs [**Bar89, BIS90**], and adapted by Clote [**Clo93**] to show that the $\mathbf{NC}^1$ functions are the least class including certain initial functions and closed under composition, concatenation recursion on notation, and K-bounded recursion on notation (for constant K).

In more detail, the initial functions of $\mathbf{ALV}'$ include all of the initial functions of **PV**, and also s and $|x|$. (The intended interpretation of s is $s(x) = x + 1$.) In general, if t is a **PV** term, and $x_1, ..., x_n$, $n \geq 0$, is a list of variables including all variables in t, then $[\lambda x_1...x_n.t]$ is an n-place function symbol. If g, h, and k are n-place, $n+1$-place, and $n+1$-place function symbols respectively, then $R^{crn}[g, h, k]$ is an $n+1$-place function symbol, representing the result of concatenation recursion on notation from g, h, and k. If g, h, and k are n-place, $n+2$-place, and $n+2$-place function symbols respectively, and $\ell > 0$, then $R_\ell[g, h, k]$ is an $n+1$-place function symbol, representing the result of K-bounded recursion on notation from g, h, and k, where $K = 2^\ell - 1$.

The axioms of $\mathbf{ALV}'$ again consist of $s_0(0) = 0$ and defining equations for each function symbol except s_0 and s_1. The rules of $\mathbf{ALV}'$ are the same as for **PV**. See Appendix A for a complete list of axioms and rules.

2.2. Interpreting $\mathbf{ALV}'$ in PV. We now give a simple interpretation of $\mathbf{ALV}'$ in **PV**. For each $\mathbf{ALV}'$ function symbol f we define an equivalent **PV** function symbol f^π with the following property: If E is a defining equation for f in $\mathbf{ALV}'$, then **PV** proves E^π, where E^π is the result of replacing each function symbol g in E by g^π. We define f^π by induction on the structure of f.

If f is an initial function of both $\mathbf{ALV}'$ and **PV**, then set $f^\pi = f$. If f is an initial function of $\mathbf{ALV}'$ but not of **PV**, then f is either s or $|\ |$. Both of these functions are easily defined in **PV**. In fact, referring to [**CU93**], s is defined in D24 and its $\mathbf{ALV}'$ defining equations follow from T26, and $|x|$ is defined in D133 and its $\mathbf{ALV}'$ defining equations follow from T135. We use these definitions to define s^π and $|\ |^\pi$.

If f is $[\lambda x_1...x_n.t]$, then we define f^π to be $[\lambda x_1...x_n.t^\pi]$, where t^π is the result of replacing each function symbol g in t by g^π. Then the transformed $\mathbf{ALV}'$ defining equation for f is precisely the **PV** defining equation for f^π, and hence it is a theorem of **PV**.

The $\mathbf{ALV}'$ function symbols $R^{crn}[g, h, k]$ and $R_\ell[g, h, k]$ represent forms of recursion on notation, and these functions can be defined using the **PV** recursor $R[g', h', k']$, for suitable g', h', and k'. In particular the bounding function k' for

$R^{crn}[g,h,k]$ is $k'(x,\vec{y}) = g(\vec{y})\star x$ (where $\star$ represents concatenation), and the bounding function for $R_\ell[g,h,k]$ is the constant $2^\ell - 1$. These bounds are provable in **PV**, and the transformed defining equations for $R^{crn}[g,h,k]$ and $R_\ell[g,h,k]$ follow.

We will identify each **ALV**$'$ function symbol f with its **PV** counterpart f^π. Thus we have proved

THEOREM 1. **PV** *is an extension of* **ALV**$'$.

2.3. The quantified theories. The systems **QPV** and **QALV** are quantified versions of **PV** and **ALV**$'$, respectively. They are first order theories whose non-logical symbols are those of **PV** (respectively **ALV**$'$) and whose axioms are the universal closures of the theorems (all of which are equations) of **PV** (respectively **ALV**$'$), together with two more:

Q1 $0 \neq 1$ (where $1 = s_1(0)$)
Q2 $\forall x(tr(x) = 0 \supset (x = 0 \vee x = 1))$

Thus we have

COROLLARY 1. **QPV** *is an extension of* **QALV**.

We emphasize that **QALV** and **QPV** are universally axiomatized theories, and we do not include any induction scheme among the axioms. However the axioms include all theorems of **ALV**$'$ and **PV**, respectively, and those theories include induction on notation as a rule. In Theorems 4 and 5 we show that suitable induction schemes are theorems of **QALV** and **QPV**.

It is obvious that **Q1** is not a logical consequence of the theorems of **ALV**$'$ and **PV**, because the trivial structure with the single element 0 forms a model of both theories. It is less obvious, but true, that **Q2** is not a logical consequence of either theory, and yet **Q1** and **Q2** do not add any new equational consequences of either theory. The next theorem states these results. We postpone the proof until Appendix B.

THEOREM 2. **Q1** *is not a logical consequence of* **ALV**$'$ *and* **Q2**, *and* **Q2** *is not a logical consequence of* **ALV**$'$*and* **Q1**. *Similarly for* **PV**. **QALV** *is a conservative extension of* **ALV**$'$, *and* **QPV** *is a conservative extension of* **PV**.

In view of this result, we should ask whether there are simple truths like **Q1** and **Q2** in the language of **QPV** which are not theorems of **QPV**. We argue below that **QPV** and **QALV** prove the notational analogs of the Peano postulates, and prove the fact that the ranges of s_0 and s_1 are disjoint and together cover $\mathbb{N}$. Later in this section we introduce $\leq$, and argue that **QPV** proves all of the Σ_1^b theorems of $\mathbf{S}_2^1$.

The tr defining equations $tr(s_0(x)) = x$ and $tr(s_1(x)) = x$, and the theorem $tr(0) = 0$ immediately imply that **QALV** (and hence **QPV**) can prove that s_0 and s_1 are one-one functions, and $x \neq 0 \supset s_0(x) \neq 0$. To prove that $s_1(x) \neq 0$ and the ranges of s_0 and s_1 are disjoint we use **Q1** and defining equations for s_0 and $parity$: $s_0(0) = 0$, $parity(s_0(x)) = 0$, and $parity(s_1(x)) = 1$.

By notational induction, **ALV**$'$ proves the following equations:

$$cond(x,0,x) = x \tag{1}$$

$$cond(x,y,z) = cond(cond(x,0,1),y,z) \tag{2}$$

$$tr(cond(x,0,1)) = 0 \tag{3}$$

$$x = cond(parity(x), s_0(tr(x)), s_1(tr(x))). \tag{4}$$

By setting $y = 0$ and $z = x$ in (2) and using (1), we see that if $cond(x,0,1) = 0$ then $x = 0$, so by (3) and **Q2**, **QALV** proves $x \neq 0 \supset cond(x,0,1) = 1$. Hence by (2) and the defining equations for *cond*,

$$\mathbf{QALV} \vdash x \neq 0 \supset cond(x,y,z) = z. \tag{5}$$

Finally, from this, (4), and the defining equations for *cond*, we have

$$\mathbf{QALV} \vdash x = s_0(tr(x)) \vee x = s_1(tr(x)). \tag{6}$$

2.4. Induction formulas.

DEFINITION 1. *An* $\mathbf{ALV}'$ *function symbol* f_A represents *(in* **QALV***) a* **QALV** *formula* A *with free variables* $x_1, ..., x_n$ *if* **QALV** *proves* $A \leftrightarrow f_A(\vec{x}) = 0$.

We wish to show that a large class of **QALV** formulas are representable in this sense. Obviously any formula of the form $t = 0$ is representable, where t is a term, and if formulas A and B are representable, then so are $\neg A$, $A \wedge B$, $A \vee B$, and $A \supset B$, by suitable use of the function *cond*.

Let us say that $x \sqsubseteq y$ (x is an initial segment of y) iff $x = trunc(y,z)$ for some z. We can define *segment quantification* by

$$\forall x \sqsubseteq y A(x) \equiv \forall z A(trunc(y,z))$$

and

$$\exists x \sqsubseteq y A(x) \equiv \exists z A(trunc(y,z)).$$

If f_A represents A, and B is $\forall x \sqsubseteq yA(x)$ or $\exists x \sqsubseteq yA(x)$, then f_B is easily defined from f_A using the recursor R_1 (1-bounded recursion on notation), and the necessary equivalences can be proved in **QALV** using induction on notation. Therefore the representable formulas are closed under segment quantification.

Define $Bit(x,y) \equiv parity(trunc(x,y))$, so $Bit(x,y)$ is the $(|y|+1)$-st bit of x from the right. Then define

$$\begin{aligned} x < y \equiv trunc(x,y) = 0 \wedge \exists z \sqsubseteq y[Bit(y,z) \neq 0 \wedge Bit(x,z) = 0 \wedge \\ \forall w \sqsubseteq y((trunc(z,w) = 0 \wedge w \neq z) \supset Bit(x,w) = Bit(y,w))]. \end{aligned}$$

Thus $x < y$ is representable. In fact the usual order properties can be proved in **QALV**, and in particular **QALV** proves $t = u \leftrightarrow \neg t < u \wedge \neg u < t$ (trichotomy). Thus every $\mathbf{ALV}'$ formula $t = u$ is representable in **QALV**.

Recall [**Bus86**] that a *sharply bounded* quantifier is one of the form $\forall x \leq |t|$ or $\exists x \leq |t|$, where t is a term. Since **QALV** proves $(\forall x \leq |y| A(x) \leftrightarrow \forall x \sqsubseteq yA(|x|))$ and $(\exists x \leq |y| A(x) \leftrightarrow \exists x \sqsubseteq yA(|x|))$ it follows that the representable formulas are closed under sharply bounded quantification. Recall that a Σ_0^b formula is one all of whose quantifiers are sharply bounded. All of the preceding paragraphs about representability clearly apply (and are well-known [**Bus86**])to **QPV** as well as **QALV**, so we have proved

THEOREM 3. *Every* Σ_0^b **QALV** *formula is representable in* **QALV**. *Every* Σ_0^b **QPV** *formula is representable in* **QPV**.

COROLLARY 2. *Every* Σ_0^b *formula of* **QALV** *represents an* $\mathbf{NC}^1$ *relation.*

Recall that **QALV** is universally axiomatized and has no induction scheme among its axioms. Nevertheless some induction formulas are theorems.

THEOREM 4 (Σ_0^b PIND Scheme). **QALV** *proves the* P*-induction formulas*

$$(A(0) \wedge \forall x(A(tr(x)) \supset A(x))) \supset \forall x A(x),$$

where $A(x)$ *is a* Σ_0^b *formula.*

Proof: Let $\vec{y}$ be a list of the variables in $A(x)$, other than x. Using concatenation recursion on notation, **QALV** can define the function F such that $F(x, \vec{y})$ is the least initial segment z of x such that $(A(tr(z)) \wedge \neg A(z)))$, and $F(x, \vec{y}) = x$ if no such z exists. Then by suitable use of induction on notation, **QALV** proves

$$(A(0) \wedge \neg A(x)) \supset (A(tr(F(x, \vec{y})) \wedge \neg A(F(x, \vec{y}))),$$

from which the P-induction formula follows. □

THEOREM 5 (Σ_0^b IND Scheme). **QPV** *proves the induction formulas*

$$(A(0) \wedge \forall x(A(x) \supset A(x+1))) \supset \forall x A(x),$$

where $A(x)$ *is a* Σ_0^b *formula.*

Proof: Using the Dowd/Statman trick, **QPV** can use binary search to define a function G such that **QPV** proves

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

from which the induction formula follows. □

We will show in the next section that **QALV** cannot prove the Σ_0^b induction formulas unless $\mathbf{P} = \mathbf{NC}^1$.

It follows from the proof of Theorem 4.7 of [**CU93**] that all of Buss's [**Bus86**] BASIC axioms for $\mathbf{S}_2^1$ are provable in **QPV**. Thus in view of Theorem 5 the arguments in chapter 6 of [**Bus86**] apply to show that **QPV** proves all of the Σ_1^b theorems of $\mathbf{S}_2^1$.

2.5. Witnessing in QALV. We now prove simple versions for **QALV** and **QPV** of Buss's witnessing theorem [**Bus86**].

THEOREM 6. *If* **QALV** *proves* $\forall\vec{x}\exists y A(\vec{x}, y)$*, where* $A(\vec{x}, y)$ *is a* Σ_0^b *formula, then there is a* **QALV** *function symbol* f *such that* **QALV** *proves* $\forall\vec{x} A(\vec{x}, f(\vec{x}))$*. The same holds when* **QPV** *replaces* **QALV** *throughout.*

Proof: By theorem 3, we need only consider the case $A(\vec{x}, y)$ is of the form $g(\vec{x}, y) = 0$. Since **QALV** is axiomatized by $\forall$-sentences, and **QALV** proves $\forall\vec{x}\exists y g(\vec{x}, y) = 0$, it follows from the Herbrand theorem that there are finitely many terms $t_1(\vec{x})$,...,$t_n(\vec{x})$ such that

$$\mathbf{QALV} \vdash \forall\vec{x}[g(\vec{x}, t_1(\vec{x})) = 0 \vee g(\vec{x}, t_2(\vec{x})) = 0 \vee ... \vee g(\vec{x}, t_n(\vec{x}) = 0].$$

We introduce a **QALV** function f such that $f(\vec{x}) = t_i(\vec{x})$, where i is the smallest index such that $g(\vec{x}, t_i(\vec{x})) = 0$. Thus the defining equation for f is

$$f(\vec{x}) = cond(g(\vec{x}, t_1(\vec{x})), t_1(\vec{x}), cond(g(\vec{x}, t_2(\vec{x})), t_2(\vec{x}), ..., \\ cond(g(\vec{x}, t_{n-1}(\vec{x})), t_{n-1}(\vec{x}), t_n(\vec{x})))...).$$

It follows easily that $\mathbf{QALV} \vdash \forall\vec{x}\ g(\vec{x}, f(\vec{x})) = 0$. □

3. The main theorem

Definition 2. *We say that* $\mathbf{ALV}'$ proves $\mathbf{P} = \mathbf{NC}^1$ *iff for every* $\mathbf{PV}$ *function symbol* f *there is an equivalent* $\mathbf{ALV}'$ *function symbol* f^ν *such that for each defining equation* A_f *for* f *in* $\mathbf{PV}$, $\mathbf{ALV}'$ *proves* A_f^ν, *where* A_f^ν *is the result of replacing every function symbol* g *in* A_f *by* g^ν. *We require that the transformation* $f \to f^\nu$ *fixes all* $\mathbf{ALV}'$ *function symbols.*

Theorem 7. *The following statements are equivalent:*

1. $\mathbf{QPV}$ *is conservative over* $\mathbf{QALV}$.
2. $\mathbf{QPV}$ *is* Σ_1^b*-conservative over* $\mathbf{QALV}$.
3. $\mathbf{QALV}$ *proves its open induction formulas.*
4. $\mathbf{ALV}'$ *proves* $\mathbf{P} = \mathbf{NC}^1$.

Proof that $1 \Longrightarrow 2$**:** Immediate. □

Proof that $2 \Longrightarrow 3$**:** Since $\mathbf{QALV}$ contains $\#$, all function symbols of $\mathbf{QALV}$ are provably bounded by terms in $\mathbf{QALV}$. In particular, $\mathbf{QPV}$ proves that the function G introduced in the proof of Theorem 5 is bounded by some $\mathbf{QALV}$ term t. Therefore $\mathbf{QPV}$ proves the Σ_1^b-formula

$$(A(0) \wedge \neg A(x)) \supset \exists z \leq t(A(z) \wedge \neg A(z+1)),$$

from which the induction formula for A follows. □

Proof that $4 \Longrightarrow 1$**:** Note that a sufficient set of axioms for $\mathbf{QPV}$ consists of the defining axioms for the $\mathbf{PV}$ function symbols, Axioms **Q1** and **Q2**, the open P-induction formulas, and the $\mathbf{QALV}$ theorem (6). Thus under the assumption 4 we can interpret $\mathbf{QPV}$ into $\mathbf{QALV}$ by replacing each function symbol f by f^ν. This translates the defining axioms for each $\mathbf{PV}$ function symbol into $\mathbf{QALV}$ theorems by assumption. Axioms **Q1** and **Q2** and the theorem (6) are fixed under the translation, and are theorems of $\mathbf{QALV}$. The open P-induction formulas remain open P-induction formulas under the translation, and are therefore theorems of $\mathbf{QALV}$ by Theorem 4. Thus theorems of $\mathbf{QPV}$ are translated into theorems of $\mathbf{QALV}$. Since $\mathbf{QALV}$ formulas are fixed under the interpretation, those that are theorems of $\mathbf{QPV}$ must also be theorems of $\mathbf{QALV}$. □

Proof that $3 \Longrightarrow 4$**:** We use the fact that a certain version of the circuit value problem is complete for $\mathbf{P}$. We consider boolean circuits C with k nodes, including m input nodes and n output nodes. We identify the nodes with the numbers $\{0, 1, ..., k-1\}$. Each node is either an *input node* or a *gate*. Each gate has one of the types AND, OR (each of fan-in two), NOT (of fan-in one), or constants 0,1 (of fan-in 0). Some gates are designated *output gates*. We assume that the nodes are reverse topologically numbered, and that nodes $0, 1, ..., n-1$ are output gates, nodes $k-m, k-m+1, ..., k-1$ are input nodes, and each gate i is less than its input node(s).

An *assignment* for such a circuit C is a map t which assigns a value $t(i)$ in $\{0,1\}$ to each node i. Similarly an *input assignment* to C is a map u which assigns 0 or 1 to each input node of C, and an *output assignment* is a map v which assigns 0 or 1 to each output gate of C. For each circuit C and input assignment u there is a unique correct assignment t and correct output assignment v, defined in the usual way.

We say that assignment t is *locally correct* at node i for circuit C and input assignment u if either i is an input node and $t(i) = u(i)$, or i is a gate and $t(i)$ is correct relative to the value(s) given by t to the input node(s) of i. Thus t is correct for C and u iff t is locally correct at all nodes of C.

We assume a suitable coding of integers for assignments and circuits via binary notation. Explicitly, we code an assignment t by the integer

$$\#(t) = \sum_{i=0}^{k-1} t(i)2^i.$$

We let t^z denote the assignment coded by z, so $t^z(i) = bit(z, i)$. Similarly u^y and v^w denote the input assignment and output assignment coded by y and w, respectively, where y is normalized to the right, so

$$y = \#(u^y) = \sum_{i=0}^{m-1} u(i + k - m)2^i.$$

We denote by C^x the circuit coded by x, where any reasonable encoding will do. Also $k(x)$, $m(x)$, and $n(x)$ are the numbers of nodes, input nodes, and output gates, respectively, of circuit C^x.

We let $t^{x,y}$ denote the correct assignment for the circuit C^x with input assignment u^y. Notice that the function $ASSIGN(x, y) = t^{x,y}(0)$ is complete for **FP**, and therefore is not an $\mathbf{NC}^1$ function, unless $\mathbf{P} = \mathbf{NC}^1$.

We now give Σ_0^b formulas for certain relations about circuits. By Corollary 2, each relation is in $\mathbf{NC}^1$.

- $lc(x, y, z, i)$ holds iff assignment t^z is locally correct at node i for circuit C^x and input assignment u^y. (The explicit formula depends on the choice of encoding x for C^x.)
- $corr(x, y, z)$ holds iff $z = \#(t^{x,y})$; i.e. z codes the correct assignment for circuit C^x and input assignment u^y.

$$corr(x, y, z) \equiv |z| \le k(x) \wedge \forall i < k(x) lc(x, y, z, i)$$

- $lesscorr(x, y, z)$ holds iff $z \le \#(t^{x,y})$.

$$\begin{aligned} &lesscorr(x, y, z) \equiv corr(x, y, z) \vee [|z| \le k(x) \wedge \\ &\quad \exists i < k(x)[bit(z, i) = 0 \wedge \neg lc(x, y, z, i) \wedge \forall j < k(x)(i < j \supset lc(x, y, z, j))]] \end{aligned}$$

By the assumption 3, **QALV** proves its open induction formulas, and thus by Theorem 3, **QALV** proves the induction formula for $lesscorr(x, y, z)$, with induction on z. Rearranging this formula we have **QALV** proves

$$\begin{aligned} &(lesscorr(x, y, 0) \wedge \neg lesscorr(x, y, z)) \supset \\ &\quad \exists w(lesscorr(x, y, w) \wedge \neg lesscorr(x, y, w + 1)). \end{aligned}$$

Let $g(x) = 2^{k(x)}$, so $g(x)$ is one more than the number of the assignment which assigns 1's to all nodes in the circuit C^x. Thus $\neg lesscorr(x, y, g(x))$ holds, and for a suitable choice of **QALV** symbol g, **QALV** proves $\neg lesscorr(x, y, g(x))$. Since also **QALV** proves $lesscorr(x, y, 0)$, it follows that **QALV** proves $\exists w(lesscorr(x, y, w) \wedge \neg lesscorr(x, y, w + 1))$. Thus by Theorem 6 there is a **QALV** function symbol F such that

(7) $\qquad \mathbf{QALV} \vdash lesscorr(x, y, F(x, y)) \wedge \neg lesscorr(x, y, F(x, y) + 1).$

If z codes the correct assignment $t^{x,y}$ then by (7) we have $F(x,y) \le z$ and not $F(x,y)+1 \le z$, so $F(x,y) = z$. We will reach the same conclusion below by an argument which does not presuppose the existence of z and which can be formalized in **QALV**, so that

$$\text{(8)} \qquad \mathbf{QALV} \vdash corr(x,y,F(x,y)).$$

In fact, setting $w = F(x,y)$, if $lesscorr(x,y,w)$ but $\neg corr(x,y,w)$ then there is a bit position i such that bits in w to the left of i are locally correct but the i-th bit of w is 0 instead of the correct value of 1. If there is a 0 bit to the right of i in w, then position i plays the same role in w and $w+1$, so that $lesscorr(x,y,w+1)$. If on the other hand all bits to the right of i in w are 1, then bit i of $w+1$ is 1 and all bits to the right of i are 0. If all these revised bits are locally correct, then $corr(x,y,w+1)$, and otherwise there is a leftmost incorrect position i', and so $lesscorr(x,y,w+1)$. Thus the fact $\neg lesscorr(x,y,w+1)$ from (7) leads to a contradiction in all cases, so we conclude $corr(x,y,w)$.

Notice that F is complete for **FP**, so it follows that $\mathbf{P} = \mathbf{NC}^1$. Our goal now is to show that $\mathbf{ALV}'$ proves $\mathbf{P} = \mathbf{NC}^1$ in the sense of definition 2. By Theorem 2 it suffices to show that **QALV** proves $\mathbf{P} = \mathbf{NC}^1$.

Every **QPV** function f is polytime and therefore is computed by some uniform polynomial size family of boolean circuits. We will define a **QALV** function symbol $circ_f$ such that $circ_f(M)$ codes a circuit which computes f when the length of each of its number arguments is at most $|M|$. Also we will define **QALV** functions in_r and out such that $in_r(M,\vec{x})$ codes the input assignment to a circuit computing a function of arity r with input $\vec{x} = (x_1,\ldots,x_r)$ with each argument at most $|M|$ bits, and $out(N,z)$ codes the output assignment of a circuit with $|N|$ output nodes and assignment t^z. Thus

$$\text{(9)} \qquad out(N,z) = z \bmod 2^{|N|}$$

and when the arity $r = 1$

$$\text{(10)} \qquad in_1(M,x) = x \bmod 2^{|M|}.$$

For each arity r we define the **QALV** input size function M_r by

$$\text{(11)} \qquad M_r(\vec{x}) = tr(2^{\max(|x_1|,\ldots,|x_r|)}).$$

For each **PV** function symbol f we define the three $\mathbf{ALV}'$ function symbols f^ν, $circ_f$, and the bound N_f, together by induction on the structure of f. Then $circ_f(M)$ represents a circuit with $|N_f(M)|$ output bits. We show that **QALV** proves the transformed defining equations for f, as required by definition 2, and also show that **QALV** proves the length monotonicity of N_f:

$$\text{(12)} \qquad |M| \le |M'| \supset |N_f(M)| \le |N_f(M')|,$$

and the bounding property of N_f:

$$\text{(13)} \qquad \bigwedge_{i=1}^{r} |x_i| \le |M| \supset |f^\nu(\vec{x})| \le |N_f(M)|,$$

and finally **QALV** proves the correctness of the circuit:

$$\text{(14)} \qquad \bigwedge_{i=1}^{r} |x_i| \le |M| \supset f^\nu(\vec{x}) = out(N_f(M), F(circ_f(M), in_r(M,\vec{x}))).$$

Since also **QALV** proves $|x_i| \le |M_r(\vec{x})|$, $1 \le i \le r$, it follows that **QALV** proves

$$f^\nu(\vec{x}) = out(N_f(M_r(\vec{x})), F(circ_f(M_r(\vec{x})), in_r(M_r(\vec{x}), \vec{x}))). \tag{15}$$

For the base case of the induction, f is an initial function symbol of **PV**. Then f is also an initial function symbol of **ALV**$'$, so we take $f^\nu = f$. The defining equations for f are the same in **ALV**$'$ and **PV**, and the transformation ν fixes these equations, so the transformed defining equations are theorems of **QALV**. The bound N_f is straightforward to define, and the circuit function $circ_f$ is either trivial, or defined using the recursive construction described below.

For the first case of the induction step, $f = [\lambda\vec{x}.t]$, where ν has been defined on the function symbols in the term t. Then we set $f^\nu = [\lambda\vec{x}.t^\nu]$, where t^ν results from t by replacing each function symbol g in t by g^ν. (If f is already an **ALV**$'$ function symbol, then t^ν is t and this identity already holds.) The transformed **PV** defining equation for f is $f^\nu(\vec{x}) = t^\nu$, which is the defining equation for f^ν in **ALV**$'$. Hence the condition for definition 2 is satisfied.

We describe the functions $circ_f$ and N_f for the special case in which t is $g(h(x))$, so $f^\nu = [\lambda x.g^\nu(h^\nu(x))]$. The general case is shown by induction on the structure of t. When t is $g(h(x))$, we set

$$N_f(M) = N_g(N_h(M)). \tag{16}$$

Then (12) and (13) follow from (12) and (13) with f replaced first by h and then by g, which hold by the induction hypothesis.

To define $circ_f$, we use C^f, C^g, and C^h to denote the circuits represented by $circ_f(M)$, $circ_g(N_h(M))$, and $circ_h(M)$, respectively (see Figure 1). Then C^f is obtained by putting together C^g and C^h_1 and identifying the output nodes of C^h_1 with the input nodes of C^g, where C^h_1 is obtained from C^h by adding $|K^g| - |N_h(M)|$ to each node number, where $|K^g|$ is the number of nodes in C^g.

To establish (14) we use the defining equation $f^\nu(\vec{x}) = g^\nu(h^\nu(\vec{x}))$ for f^ν, and also (14) with f replaced first by h and then by g, and (13) with f replaced by h. These hold by the induction hypothesis. Since the nodes of C_g have been uniformly renumbered, we also need the fact that F behaves similarly on a circuit and its renumbered version. This is shown by the correctness (8) of F.

For the remaining case, $f = R[g, h, k]$, and f is defined from g, h by recursion on notation, with bounding function k. The defining equation for f is

$$f(x, \vec{y}) = cond(x, g(\vec{y}), cond(trunc(t, k(x, \vec{y})), t, k(x, \vec{y}))), \tag{17}$$

where $t \equiv h(x, \vec{y}, f(tr(x), \vec{y}))$.

We introduce three auxiliary functions FF, H and K by the defining equations

$$H(x, \vec{y}, z) = cond(x, g(\vec{y}), h(x, \vec{y}, z)), \tag{18}$$

$$K(x, \vec{y}) = cond(x, g(\vec{y}), k(x, \vec{y})), \tag{19}$$

and

$$FF(x, \vec{y}, z) = cond(trunc(H(x, \vec{y}, z), K(x, \vec{y})), H(x, \vec{y}, z), K(x, \vec{y}))). \tag{20}$$

If follows immediately from (17) and these definitions that

$$f(x, \vec{y}) = FF(x, \vec{y}, f(tr(x), \vec{y})). \tag{21}$$

FIGURE 1. Circuit C^f for $f(x) = g(h(x))$.

We may assume that the induction hypothesis applies to FF, H, and K by the previous case in our induction. Thus let

$$N_f(M) = N_K(M). \tag{22}$$

We base the definition of the circuit $C^f = circ_f(M)$ on the recursion (21). Thus let

$$M' = \max(M, N_K(M)) \tag{23}$$

and

$$C^{FF} = circ_{FF}(M') \tag{24}$$

Then C_f consists of a stack of $|M| + 1$ copies of C^{FF}, with inputs and outputs suitably directed and nodes suitably renumbered (see Figure 2). We designate these copies $C^{FF}_{|M|}, C^{FF}_{|M|-1}, ..., C^{FF}_0$. The inputs to C^f are $(x, \vec{y})$ and the inputs to C^{FF} are $(x', \vec{y}, z')$. for suitable x' and y'. The inputs $(x', \vec{y}, z')$ of the top copy $C^{FF}_{|M|}$ are set to $(0, \vec{y}, 0)$. For $0 \leq j < |M| - 1$ the inputs to copy C^{FF}_j are $(\lfloor x/2^j \rfloor, \vec{y}, w)$ where w is the output from the copy above. The output to C^f is the output to copy C^{FF}_0, padded with zeroes or chopped to exactly $|N_K(M)|$ bits.

FIGURE 2. C^f for $f(x, \vec{y}) = FF(x, \vec{y}, f(t_r(x), \vec{y}))$.

Each input number for C^f has $|M|$ bits, but each input number for C^{FF} has $|M'|$ bits. Because $|M'|$ may exceed $|M|$, the high order bits of some input arguments of the C^{FF} copies may have to be set to 0. (In Figure 2 we assume $M = M'$.) Also, each copy of C^{FF} has $|N_{FF}(M')|$ output bits. If $|N_{FF}(M')| > |M'|$, then only the $|M'|$ low-order output bits of C^{FF} are connected to he z-input of the copy below. (The discarded output bits are always 0.)

We now calculate the total number k_f of nodes in C^f and the amount added to each node number of each copy of C^{FF}. Assuming that $\vec{y}$ has arity r, the total number of input bits to C^f to code the input $(x, \vec{y})$ is $m_f = |M|(r+1)$. By convention, these bits have node numbers $k_f - m_f, k_f - m_f + 1, ..., k_f - 1$. For simplicity, we mandate that for each copy of C^{FF}, each input bit set to 0 is identified with a single 0 gate, numbered $k_f - m_f - 1$. Thus each of the $m_{FF} = |M'|(r+2)$ input bits to each copy of C^{FF} is identified with this 0 gate, or with an input to C^f, or with an output to the copy of C^{FF} above. Also the bottom copy C_0^{FF} must have $B = \max(|N_{FF}(M)| - |N_K(M)|, 0)$ constant 0 output gates inserted to pad its output to $|N_K(M)|$ bits. Therefore $k_f = B + (|M|+1)(k_{FF} - m_{FF}) + m_f + 1$, where k_{FF} is the number of nodes in C^{FF}. For $0 \leq j \leq |M|$, copy C_j^{FF} has each of its non-input node numbers incremented by $B + j(k_{FF} - m_{FF})$, except that the output gates of the bottom copy C_0^{FF} are unchanged.

This completes our description of the circuit function $circ_f(M)$. We point out that the length multiplications needed to renumber gates are readily implemented in **QALV** using the # function.

If f is not already an **ALV**$'$ function symbol, we define

$$f^\nu = [\lambda x\vec{y}.T], \tag{25}$$

where T is the right hand side of (15) (with $(\vec{x})$ replaced by $(x, \vec{y})$).

If f is already an **ALV**$'$ function symbol, then we must take $f^\nu = f$. In this case $f = f^\pi$ in the interpretation defined before Theorem 1, and so f is defined by concatenation recursion on notation or K-bounded recursion on notation, and its defining equations in **ALV**$'$ and **PV** are similar and inter-provable in **QALV**. It will follow from the argument below that **QALV** proves $f(x, \vec{y}) = T$, where T is as in (25). Therefore we will not further consider this case separately.

Monotonicity (12) of N_f follows immediately from (22) and the induction hypothesis (12) for N_K. To prove (13), we can show in **QALV**

$$\bigwedge_{i=1}^{r} |x_i| \le |M| \supset |M_r(\vec{x})| \le |M|.$$

Then (13) follows easily from (15), (12), and the simple fact $|out(N, z)| \le |N|$ from (9).

To show that **QALV** proves the circuit correctness (14) we must compare the results of two circuits for f: $C^f = circ_f(M)$ and the circuit $\hat{C}^f = circ_f(M_{r+1}(x, \vec{y}))$ which computes f^ν according to (15). For this we need to know that the copies of C^{FF} in the two circuits each correctly compute FF^ν, and we need properties of FF^ν. We can use the transformed defining equations (18^ν), (19^ν), and (20^ν) for H^ν, K^ν, and FF^ν, which follow from (18), (19), and (20) by the induction hypothesis. It follows easily from the transformed equations that

$$|FF^\nu(x, \vec{y}, z)| \le |K^\nu(x, \vec{y})|. \tag{26}$$

The correctness of each copy of C^{FF} follows from (14), with f replaced by FF, which holds by the induction hypothesis. We also need to know that the outputs of these circuits are not too big, since we only use the $|M'|$ low-order bits of the output as inputs to the copy below. Since we are trying to prove (14) for f, we may assume that each of the argument $(x, y_1, ..., y_r)$ is bounded in length by $|M|$, so by (26) we have

$$|FF^\nu(\lfloor x/2^j \rfloor, \vec{y}, z)| \le |N_K(M)|,$$

where we have used the bound (13) with f replaced by K. This bound suffices, since we are allowing $|M'| \ge |N_K(M)|$ input bits for each number input to C^{FF} in C^f. A similar argument applies to $\hat{C}^f$, with M replaced by $M_r(\vec{x})$.

Now from the correctness of the circuits C^{FF}, we can show that among the bottom $|x|$ copies of C^{FF} in the two circuits C^f and $\hat{C}^f$, corresponding copies compute the same values. From the defining equations, we conclude that $FF^\nu(0, \vec{y}, z) = g^\nu(\vec{y})$, so the output of copy $C^{FF}_{|x|}$ in the two circuits is the same; namely $g^\nu(\vec{y})$ (see Figure 2). Proceeding by P-induction on w for $|w| \le |x|$, the output of copy $C^{FF}_{|trunc(x,w)|}$ in C^f represents the same number as the output of copy $\hat{C}^{FF}_{|trunc(x,w)|}$ in $\hat{C}^f$. In particular, for $w = x$, we conclude that the outputs of the two circuits C^f and $\hat{C}^f$ are the same. This established (14).

It remains to show that **QALV** proves the transformed defining equation (17^ν) for f^ν. This is easily equivalent to establishing (21^ν) in **QALV**. ¿From (14) we see that the circuit $C^f = circ_f(M)$, with $M = M_{r+1}(x, \vec{y})$, correctly computes both $f^\nu(x, \vec{y})$ and $f^\nu(tr(x), \vec{y})$ (see Figure 2). Arguing as above, we can compare the outputs of the bottom $|x|$ copies of C^{FF} in C^f with input $(x, \vec{y})$ and input $(tr(x), \vec{y})$. We see by P-induction that for $0 \le j \le |tr(x)|$, the output of copy C_j^{FF} for C^f-input $(tr(x), \vec{y})$ is the same as the output of copy C_{j+1}^{FF} for C^f-input $(x, \vec{y})$. In particular, for $j = 0$, the output of C^f with input $(tr(x), \vec{y})$ (namely $f^\nu(tr(x), \vec{y})$) is the same as the output of the second-from-bottom copy C_1^{FF} of C^f with input $(x, \vec{y})$. Application of the bottom copy C_0^{FF} gives $f^\nu(x, \vec{y})$, and (21^ν) follows. $\square$

4. Discussion

As mentioned in the Introduction, one motivation for this paper is to relate the question $\mathbf{P} = \mathbf{NC}^1$ to the question of whether Frege systems can p-simulate extended Frege systems. It is natural to try to find combinatorial principles which separate Frege and extended Frege systems, based on the perception that their proofs require polynomial time concepts which are not in $\mathbf{NC}^1$. Analogous principles are known for the problem of separating Frege and constant-depth Frege systems. For example, the pigeonhole principle, formulated as a sequence of tautologies, has polynomial size Frege proofs but requires exponential size constant-depth Frege proofs [**Kra95a**].

In [**BBP95**], Bonet, Buss, and Pitassi suggest several combinatorial principles which may separate Frege and extended Frege systems. Some of these, including the Odd-town theorem, seem to require linear algebra for their proofs. Linear algebra proofs usually depend on some form of Gaussian elimination, which is not known to be expressible using $\mathbf{NC}^1$ functions. Thus no one knows how to formalize the proofs of these combinatorial principles in **ALV**$'$ (or **AID** or $\mathbf{T}_1$), although they are readily formalized in **PV**. Thus we have no techniques for constructing polynomial size proofs of the corresponding tautologies.

Incidently, a principle whose subject matter is linear algebra is the following: If A and B are $n \times n$ matrices over GF(2) such that $AB = I$, then $BA = I$. The proof seems to require Gaussian elimination and can be formalized in **PV**, but, as far as we know, not in **ALV**$'$.

Paraphrasing the Assertion in the Introduction, it seems likely that one can show that **PV** is conservative over **ALV**$'$ iff **ALV**$'$ proves that Frege systems p-simulate extended Frege systems. It follows from the easy part of Theorem 7 that if **ALV**$'$ proves that $\mathbf{P} = \mathbf{NC}^1$ then **PV** is conservative over **ALV**$'$. A major problem is to strengthen Theorem 7 and show a converse: If **PV** is conservative over **ALV**$'$ (i.e. **ALV**$'$ proves that Frege systems p-simulate extended Frege systems) then $\mathbf{P} = \mathbf{NC}^1$.

Appendix A. Axioms and Rules of QALV and QPV

A.1. Axioms for PV.

A1. $s_0(0) = 0$
A2. $parity(s_0(x)) = 0$
A3. $parity(s_1(x)) = 1$ $(= s_1(0))$
A4. $tr(s_0(x)) = x$
A5. $tr(s_1(x)) = x$

A6. $cond(0, y, z) = y$
A7. $cond(s_0(x), y, z) = cond(x, y, z)$
A8. $cond(s_1(x), y, z) = z$
A9. $pad(x, s_0(y)) = cond(y, x, s_0(pad(x, y))$
A10. $pad(x, s_1(y)) = s_0(pad(x, y))$
A11. $trunc(x, s_0(y)) = cond(y, x, tr(trunc(x, y)))$
A12. $trunc(x, s_1(y)) = tr(trunc(x, y))$
A13. $x\#s_0(y) = cond(y, 1, pad(x\#y, x))$
A14. $x\#s_1(y) = pad(x\#y, x)$
A15. $[\lambda x_1 ... x_n.t](x_1, ..., x_n) = t$

BRN. $R[g, h, k](x, \vec{y}) = cond(x, g(\vec{y}), cond(trunc(t, k(x, \vec{y})), t, k(x, \vec{y})))$, where
$t \equiv h(x, \vec{y}, R[g, h, k](tr(x), \vec{y}))$

A.2. Axioms for ALV′.

A1-A15, and also
A16. $s(0) = 1$
A17. $s(s_0(x)) = s_1(x)$
A18. $s(s_1(x)) = s_0(s(x))$
A19. $|0| = 0$
A20. $|s_0(x)| = cond(x, 0, s(|x|))$
A21. $|s_1(x)| = s(|x|)$

CRN. $R^{crn}[g, h, k](x, \vec{y}) = cond(x, g(\vec{y}), cond(parity(x), U, V))$, where
$U = cond(h(tr(x), \vec{y}), s_0(t), s_1(t))$ and
$V = cond(k(tr(x), \vec{y}), s_0(t), s_1(t))$ and
$t = R^{crn}[g, h, k](tr(x), \vec{y})$.
KBRN. $R_\ell[g, h, k](x, \vec{y}) = cond(x, g(\vec{y}), cond(parity(x), U_\ell(x, \vec{y}), V_\ell(x, \vec{y})))$, where
$U_\ell(x, \vec{y}) = h(tr(x), \vec{y}, R_\ell[g, h, k](tr(x), \vec{y})) \bmod 2^\ell$ and
$V_\ell(x, \vec{y}) = k(tr(x), \vec{y}, R_\ell[g, h, k](tr(x), \vec{y})) \bmod 2^\ell$.
Here $x \bmod 2^1$ is $parity(x)$, $x \bmod 2^2$ is
$cond(parity(x), s_0(parity(tr(x))), s_1(parity(tr(x))))$, etc.

A.3. Rules of ALV′ and PV.

R1. $t = u \vdash u = t$
R2. $t = u, u = v, \vdash t = v$
R3. $t_1 = u_1, ..., t_n = u_n \vdash f(t_1, ..., t_n) = f(u_1, ..., u_n)$
R4. $t = u \vdash t[v/x] = u[v/x]$, x a variable, v any term.
R5. Induction on notation:

$$\frac{\begin{array}{c} t_1[0/x] = t_2[0/x] \\ t_1[s_0(x)/x] = v_0[t_1/a] \quad t_2[s_0(x)/x] = v_0[t_2/a] \\ t_1[s_1(x)/x] = v_1[t_1/a] \quad t_2[s_1(x)/x] = v_1[t_2/a] \end{array}}{t_1 = t_2}$$

Appendix B. Proof of Theorem 2

As mentioned before Theorem 2, the single element universe $\{0\}$ forms a model for **ALV′**, **PV**, and **Q2**, and hence $0 \neq 1$ is not a logical consequence of **Q2** together with the theorems of these theories.

To prove the rest of Theorem 2, we need a simple property of Horn clause sets. Recall that a Horn clause is a disjunction of literals, where each literal is a negated atom, except we allow at most one unnegated atom per clause. The property we need is that every propositionally consistent set of Horn clauses has a unique *minimal model*; that is, a truth assignment τ which satisfies all clauses in the set, and further for each atom P, if P^τ = TRUE, then P^σ = TRUE for every assignment σ which satisfies the set. The minimal model is easily constructed iteratively, starting with the totally FALSE assignment.

THEOREM 8. **Q2** *is not a logical consequence of* **Q1** *and the theorems of* $\mathbf{ALV}'$*, and not a logical consequence of* **Q1** *and the theorems of* **PV***.*

Proof: We give the argument for **PV**. The argument for $\mathbf{ALV}'$ is the same.

Let $\mathbf{PV}^0$ denote the set of all theorems of **PV**, together with the set of all substitution instances of the standard equality axioms for the language of **PV**. Then $\mathbf{PV}^0$ is a set of Horn clauses. Suppose that **Q2** is a logical consequence of the theorems of **PV** together with $0 \neq 1$. Then by the Herbrand Theorem, **Q2** without its quantifier, i.e. $tr(x) = 0 \supset (x = 0 \vee x = 1)$, is a truth functional consequence of $\mathbf{PV}^0 \cup \{0 \neq 1\}$. Since $\mathbf{PV}^0 \cup \{0 \neq 1, tr(x) = 0\}$ is a consistent set of Horn clauses, it has has a minimal model τ. Then $(x = 0 \vee x = 1)^\tau$ = TRUE. But $(x = 0)^\tau$ = FALSE and $(x = 1)^\tau$ = FALSE, because τ is minimal and each of the inequations $x \neq 0$ and $x \neq 1$ can be added separately to $\mathbf{PV}^0 \cup \{0 \neq 1, tr(x) = 0\}$ to form a consistent set. This is a contradiction. □

THEOREM 9. **QALV** *is a conservative extension of* $\mathbf{ALV}'$ *and* **QPV** *is a conservative extension of* **PV***.*

Proof: Again we prove this for the case of **PV**. We need certain theorems of **PV**, but the **PV** proofs of these involve only the initial functions 0, s_0, s_1, tr, and $cond$, so these theorems are also theorems of $\mathbf{ALV}'$.

Suppose $u = v$ is a logical consequence of the theorems of **PV**, together with **Q1** and **Q2**. Then by the Herbrand Theorem, there is a finite set of substitution instances $A_1, ..., A_n$ of **Q2** such that $u = v$ is a truth functional consequence of $\mathbf{PV}^0 \cup \{0 \neq 1, A_1, ..., A_n\}$, where $\mathbf{PV}^0$ is as defined in the proof of Theorem 8. By Lemma 5 below, $u = v$ is a consequence of $\mathbf{PV}^0$, so $\mathbf{PV} \vdash u = v$. (We use the fact that the equality rules of **PV** are complete for equational first order consequences.) □

For the rest of the proof, **E** denotes an arbitrary set of **PV** equations, and $\models$ denotes propositional logical consequence.

¿From the defining equations for $cond$, we have that **PV** proves $cond(0, u, v) = u$ and $cond(1, u, v) = v$, and therefore

$$\mathbf{PV}^0, 0 = 1 \models u = v \tag{27}$$

for any **PV** terms u and v. Thus we can prove

LEMMA 1. *If* $\mathbf{PV}^0, \mathbf{E}, 0 \neq 1 \models u = v$*, then* $\mathbf{PV}^0, \mathbf{E} \models u = v$ *for any* **PV** *terms* u *and* v*.*

Proof: If $\mathbf{PV}^0 \cup \mathbf{E}$ is inconsistent the result is obvious. Otherwise let τ be the minimal model for $\mathbf{PV}^0 \cup \mathbf{E}$. If $(0 = 1)^\tau$ = FALSE, then by hypothesis $(u = v)^\tau$ = TRUE, so $(u = v)^\sigma$ = TRUE for every assignment σ satisfying $\mathbf{PV}^0 \cup \mathbf{E}$. If $(0 = 1)^\tau$ = TRUE then $\mathbf{PV}^0 \cup \mathbf{E} \models 0 = 1$, so the result follows from (27). □

LEMMA 2. *If* $\mathbf{PV}^0, \mathbf{E}, t = 0 \models u = v$ *then* $\mathbf{PV}^0, \mathbf{E} \models cond(t, u, v) = v$*, for any* $\mathbf{PV}$ *terms* t*,* u*, and* v*.*

Proof: If $\mathbf{PV}^0, \mathbf{E}, t = 0 \models u = v$ then $u = v$ can be derived from theorems of $\mathbf{PV}$ $\mathbf{E}$, and $t = 0$ by repeated use of the equality rules **R1, R2**, and **R3**. The Lemma is proved by induction on the length of this derivation. The first part of the base case is when $u = v$ is either a theorem of $\mathbf{PV}$ or in $\mathbf{E}$. By notational induction on x, $\mathbf{PV}$ proves $cond(x, y, y) = y$, and hence $\mathbf{PV}$ proves $cond(t, v, v) = v$. By the equality axioms and $u = v$, we conclude $\mathbf{PV}^0, \mathbf{E} \models cond(t, u, v) = v$. The second part of the base case is when $u = v$ is $t = 0$. By notational induction on x, $\mathbf{PV}$ proves $cond(x, x, 0) = 0$ so $\mathbf{PV}$ proves $cond(t, t, 0) = 0$.

For the first part of the induction step, suppose $u = v$ results from the application of rule **R1** to $v = u$. By the induction hypothesis, $\mathbf{PV}^0, \mathbf{E} \models cond(t, v, u) = u$. Using appropriate substitution instances of the $\mathbf{PV}$ theorems $cond(x, y, y) = y$ and

$$cond(x, cond(x, y, z), w) = cond(x, y, w)$$

(see **T4**, p123 of [**CU93**]), by the equality axioms we conclude

$$\begin{aligned} &(28) & \mathbf{PV}^0, \mathbf{E} \models cond(t, u, v) &= cond(t, cond(t, v, u), v) \\ &(29) & &= cond(t, v, v) = v. \end{aligned}$$

The other two rules are handled similarly. (The reasoning is similar to the justification of **DR13** and **DR14** of [**CU93**].) □

LEMMA 3. *If* $\mathbf{PV}^0, \mathbf{E}, t = 1 \models u = v$ *then* $\mathbf{PV}^0, \mathbf{E}, tr(t) = 0 \models cond(t, u, v) = u$*, for any* $\mathbf{PV}$ *terms* t*,* u*, and* v*.*

Proof: As in the proof of the above lemma, we proceed by induction on the derivation of $u = v$ via **R1, R2**, and **R3**. The induction step is the same as the justification of **DR12, DR13**, and **DR14** in [**CU93**]. The interesting part of the base case is when $u = v$ is $t = 1$. For this we need the $\mathbf{PV}$ theorem

$$cond(tr(x), cond(x, 1, x), 1) = 1. \tag{30}$$

This can be proven by Conditional Induction on x (**DR17** of [**CU93**]), with a secondary induction to take care of the induction step $s_1(x)/x$. From (30) we conclude $\mathbf{PV}^0, tr(t) = 0 \models cond(t, 1, t) = 1$, as required. □

LEMMA 4. *If* $\mathbf{PV}^0, \mathbf{E} \models tr(t) = 0$ *and* $\mathbf{PV}^0, \mathbf{E}, t = 0 \models u = v$ *and* $\mathbf{PV}^0, \mathbf{E}, t = 1 \models u = v$*, then* $\mathbf{PV}^0, \mathbf{E} \models u = v$*.*

Proof: Immediate from the two preceding lemmas. □

LEMMA 5. *Let* $A_1, ..., A_n$ *be* n *substitution instances of* **Q2**. *If*

$$\mathbf{PV}^0, \mathbf{E}, 0 \neq 1, A_1, ..., A_n \models u = v, \tag{31}$$

then $\mathbf{PV}^0, \mathbf{E} \models u = v$*.*

Proof: Induction on n. If $\mathbf{PV}^0 \cup \mathbf{E} \cup \{0 \neq 1\}$ is inconsistent, we are done by Lemma 1, so let τ be the minimal model for $\mathbf{PV}^0 \cup \mathbf{E} \cup \{0 \neq 1\}$. Let A_i be $tr_i(t_i) = 0 \supset (t_i = 0 \vee t_i = 1)$, for $i = 1, ..., n$, and assume the hypothesis (31). If $(tr(t_i) = 0)^\tau =$ FALSE for all i, then $(u = v)^\tau =$ TRUE, and we are done by Lemma 1. Otherwise we may assume $(tr(t_1) = 0)^\tau =$ TRUE, so $\mathbf{PV}^0, \mathbf{E} \models tr(t_1) = 0$ by Lemma 1. Also, by (31),

$$\mathbf{PV}^0, \mathbf{E}, t_1 = 0, 0 \neq 1, A_2, ..., A_n \models u = v$$

so $\mathbf{PV}^0, \mathbf{E}, t_1 = 0 \models u = v$ by the induction hypothesis. Similarly $\mathbf{PV}^0, \mathbf{E}, t_1 = 1 \models u = v$. Therefore $\mathbf{PV}^0, \mathbf{E} \models u = v$ by Lemma 4. $\square$

Acknowledgment: I thank Sam Buss and especially Jan Krajíček for their illuminating comments on the first version of this paper.

References

[Ara91] T. Arai, *Frege systems, ALOGTIME and bounded arithmetic*, 1991, Manuscript.

[Bar89] D. A. Barrington, *Bounded-width polynomial-time branching programs recognize exactly those languages in NC^1*, J. Comp. Systems Sci. **38** (1989), no. 1, 150–164.

[BBP95] M. L. Bonet, S. R. Buss, and T. Pitassi, *Are there hard examples for Frege systems?*, Feasible Mathematics II (P. Clote and J.B. Remmel, eds.), Birkhauser, 1995, pp. 30–56.

[BIS90] D. M. Barrington, N. Immerman, and H. Straubing, *On uniformity in NC^1*, J. Comp. Systems Sci. **41** (1990), no. 3, 274–306.

[Bus86] S. R. Buss, *Bounded Arithmetic*, 1986, Naples, Bibliopolis. (Revision of 1985 Princeton University Ph.D. thesis).

[Bus91] S. R. Buss, *Propositional consistency proofs*, Annals of Pure and Applied Logic **52** (1991), 3–29.

[Bus95] S. R. Buss, *Relating the bounded arithmetic and polynomial-time hierarchies*, Annals of Pure and Applied Logic **75** (1995), 67–77.

[Clo90] P. Clote, *Alogtime and a conjecture of s.a. cook*, Proceedings of IEEE Symposium on Logic in Computer Science, 1990.

[Clo93] P. Clote, *On polynomial size Frege proofs of certain combinatiorial principles*, Arithmetic, Proof Theory, and Computational Complexity (P. Clote and J. Krajiček, eds.), Oxford, 1993, pp. 162–184.

[Cob65] A. Cobham, *The intrinsic computational difficulty of functions*, Proceedings of the International Congress Logic, Methodology, and Philosophy of Science (Y. Bar-Hillel, ed.), North Holland, 1965, pp. 24–30.

[Coo75] S. A. Cook, *Feasibly constructive proofs and the propositional calculus*, Proceedings of the 7th Annual ACM Symposium on the Theory of Computing, 1975, pp. 83–97.

[CR79] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*, J. Symbolic Logic **44** (1979), no. 1, 36–50.

[CU93] S. A. Cook and A. Urquhart, *Functional interpretations of feasibly constructive arithmetic*, Annals of Pure and Applied Logic **63** (1993), 103–200.

[KPT91] J. Krajiček, P. Pudlak, and G. Takeuti, *Bounded arithmetic and the polynomial hierarchy*, Annals of Pure and Applied Logic **52** (1991), 143–153.

[Kra95a] J. Krajiček, *Bounded arithmetic, propositional logic, and complexity theory*, Cambridge University Press, 1995.

[Kra95b] J. Krajíček, *On Frege and extended Frege proof systems*, Feasible Mathematics II (P. Clote and J. Remmel, eds.), Birkhauser, 1995, pp. 284–319.

[Pit96] F. Pitt, *A quantifier-free theory based on a string algebra for NC^1*, Submitted, 1996.

[Ruz81] W. L. Ruzzo, *On uniform circuit complexity*, J. Comp. and Systems Sci. **22** (1981), no. 3, 365–383.

[Zam96] D. Zambella, *Notes on polynomially bounded arithmetic*, J. Symbolic Logic **61** (1996), no. 3, 942–966.

Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4

E-mail address: sacook@cs.toronto.edu