

Bounded Arithmetic

qedsphere

January 16, 2026

Contents

1	Proof Complexity Generators	2
1.1	Logic preliminaries	2
1.1.1	Cobham's limited recursion	2
1.1.2	Cook's Theory PV	3
1.1.3	Buss's S_1^2	4
1.2	Computational Complexity Preliminaries	5

1 Proof Complexity Generators

Definition 1.1 (dWPHP(f)). $\exists y < 2a \forall x < a f(x) \neq y$

Problem 1.1 (Crux). Is BT Σ_1^b -conservative over S_2^1 ?

Note these basic definitions.

Definition 1.2 (Conservativity). In this case, BT must prove all Σ_1^b statements that S_2^1 can prove (trivial; BT is defined as $S_2^1 + dWPHP(\Delta_1^b)$). Furthermore BT cannot prove any Σ_1^b statements that S_2^1 cannot prove.

A dual question, likely easier:

Problem 1.2. Does S_2^1 prove the dWPHP for all p-time functions, i.e. $S_2^1 = BT$?

Note that the quantifiers of instances of $dWPHP(\Delta_1^b)$ are of $\forall\exists\forall$. However to show nonconservativity we consider individual instances with $\exists\forall$ quantifiers that Σ_1^b cannot prove. The first statement is more specific, therefore, perhaps easier to show.

1.1 Logic preliminaries

1.1.1 Cobham's limited recursion

Cook's theory PV deals with equational statements.

Definition 1.3 (FP). FP is the set of functions $f(x_1, \dots, x_k) : (\{0, 1\}^k \rightarrow \{0, 1\}^*)$ computable by poly-time algorithms.

An example is the function that returns whether the input is a prime number.

Now a bunch of definitions:

- $c(x) = \epsilon$
- $\circ(x, y)$ concatenates x, y
- $s_i(x) = x \circ i$ concatenates a single digit
- $\#(x, y)$ repeats x $|y|$ many times.
- $TR(x)$ truncates x .
- $\pi_i(x_1, \dots, x_k) = x_i$ takes one coordinate of x .

Cobham defines two rules for defining new functions from existing functions:

- Composition: create $h(g_1(\vec{y}), \dots, g_k(\vec{y}))$.
- Limited Recursion: Base case $g(\vec{x})$, recursive case with $h_i(\vec{x}, y, z)$ handling different cases of i and defined as $f(\vec{x}, s_i(y)) = h_i(\vec{x}, y, f(\vec{x}, y)), i \in \{0, 1\}$. It limited by that $|f(\vec{x}, y)| \leq |k(\vec{x}, y)|$ for some existing function $k(\vec{x}, y)$.

The smallest class of functions that is closed under this actually happens to equal FP.

1.1.2 Cook's Theory PV

Note that $|f(\vec{x}, y)| \leq |k(\vec{x}, y)|$ must be a statement still within the theory. How is this possible? We do not only construct functions within poly time, but also their proofs.

To build PV, we first require a different limit on length: $|h_i(\vec{x}, y, z)| \leq |z \circ k_i(\vec{x}, y)|$ for $i \in \{0, 1\}$. This is at least as strong as the Cobham thing, as we can recurse over index i to build a single, adequate k function. (Actually they're the same limit.)

Now define PV's base case, or order 0 PV:

- ϵ is an empty string.
- We have $s_0(x)$, $s_1(x)$, $\circ(x, y)$, $\#(x, y)$, and $\text{TR}(x, y)$
- Additionally we define $\text{ITR}(x, y)$ which removes the leftmost $|y|$ bits of x .
- **Terms** of order i are compositions of order i functions and others.
- **Equations** of order i equate terms of order i . " $s = t$ ".

Here are some axioms to accompany our order 0 functions.

- $x \circ \epsilon = x$, $x \circ s_i(y) = s_i(x \circ y)$
- $x \# \epsilon = \epsilon$, bla, bla bla. More very intuitive axioms, two each, for TR, ITR.

Finally we define how to introduce new functions:

Definition 1.4 (Function introduction rules). • *Composition: From order- $i - 1$ term t with variables \vec{x} , create order- i function $f_t^{(0)}(\vec{x})$.*

- *Recursion: Given order- $i - 1$ proofs π_i of the equation $\text{ITR}(h_i(\vec{x}, y, z), z \circ k_i(\vec{x}, y)) = \epsilon$ mimicking our previous limits on length, create $f_{\Pi := (g, h_0, h_1, k_0, k_1, \pi_0, \pi_1)}^{(1)}$ and the axioms*

$$\begin{aligned} f_{\Pi}^{(1)}(\vec{x}, 0) &= g(\vec{x}) \\ f_{\Pi}^{(1)}(\vec{x}, s_i(y)) &= h_i(\vec{x}, y, f_{\Pi}^{(1)}) \end{aligned}$$

With their proofs:

Definition 1.5 (Proofs). *An order- i proof is a sequence of order- i equations (e_1, \dots, e_l) of the form $e_l = "s = t"$.*

To write proofs, we use logic:

- *We are given reflexivity, transitivity, and commutativity of equivalence.*
- *If $s_i = t_i$ have all been introduced then $f(s_1, \dots, s_n) = f(t_1, \dots, t_n)$.*
- *If $s = t$ has been introduced then $s[x/v] = t[x/v]$ can be introduced.*
- *Definition axioms of order- i functions may be freely introduced.*
- *Finally, induction: If we have*

$$\begin{aligned} f_1(\vec{x}, \epsilon) &= g(\vec{x}), \quad f_1(\vec{x}, s_i(y)) = h_0(\vec{x}, y, f_1(\vec{x}, y)) \\ f_1(\vec{x}, \epsilon) &= g(\vec{x}), \quad f_1(\vec{x}, s_i(y)) = h_0(\vec{x}, y, f_2(\vec{x}, y)) \end{aligned}$$

then $f_1(\vec{x}, y) = f_2(\vec{x}, y)$.

Remark 1.1. *PV characterizes both “polynomially verifiable” as well as “feasible mathematics”, both informal proposals for desirable qualities, and P .*

Definition 1.6 ($S_2^1(PV)$). *Note that S_2^1 also captures P , but it does this using only a few basic function symbols. Therefore we conservatively enrich S_2^1 by adding the symbols of all “clocked polynomial-time functions” and the corresponding first-order (PV_1) axioms, to make our lives easier.*

Remark 1.2. *If $NP \subset P/poly$, then PV_1 and $S_2^1(PV)$ are the same.*

Thus the main result of this section is that, since PV captures poly-time functions, we can rewrite:

Problem 1.3. *Does $S_2^1(PV)$ prove $dWPHP(PV)$? I.e. does it prove all formulas $dWPHP(f)$ for all function symbols f in our language?*

This problem is also open for PV_1 .

1.1.3 Buss’s S_1^2

Search up what bounded and sharply bounded quantifiers are, if you don’t know.

Definition 1.7 (Quantifier alternation classes). *Note the following.*

- $\Delta_0^b = \Sigma_0^b = \Pi_0^b$ is the set of sharply bounded formulas, and are P .
- Σ_{i+1}^b is the closure of Π_i^b under \wedge, \vee , sharply bounded quantifiers, and bounded existential quantifiers.
- Π_i^b is the closure of Σ_{i+1}^b under \wedge, \vee , sharply bounded quantifiers, and bounded universal quantifiers.

Definition 1.8 (S_2^1). *BASIC is a set of axioms defining desired properties of our $\wedge, \vee, \#$, etc. P -induction is a set of axioms acting on statements in a class of formulas. For $A \in \phi$, it says*

$$A(0) \wedge (\forall x)(A(\lfloor \frac{1}{2}x \rfloor) \rightarrow A(x)) \rightarrow (\forall x)A(x)$$

representing polynomially feasible induction.

Then S_2^i is the set of axioms BASIC + Σ_i^b -PIND.

Theorem 1.1 (Main Theorem for S_2^1). *The set of Σ_1^b definable functions provable from S_2^1 is the set of problems in P .*

Proof. This is proven using sequent calculus and witnessing lemma. It is summarized here. We will discuss such proofs soon. \square

First, we return to a motivating (open) question:

Problem 1.4. *Is full bounded arithmetic finitely axiomatizable? In particular, is $S_2^1 = S_2$? (Where S_2 is defined as the union of all S_2^i ’s.)*

Theorem 1.2. $BT = S_2^1 + dWPHP(\Delta_1^b)$ is finitely axiomatized by the instance of $dWPHP$ on the circuit value function $CV(x, y)$.

It is good to know the following principle $dWPHP_1(f, g)$:

Definition 1.9 ($dWPHP_1(f, g)$). $\exists y < 2a \ g(y) \geq a \vee f(g(y)) \neq y$

Skip forward a bit in PCG, and we have the following conservativity question relationships:

$$S_2^1(PV) \preceq_{\Sigma_1^b} BT \Leftrightarrow PV_1 \preceq_{\Sigma_1^b} BT \Leftrightarrow PV_1 \preceq_{\Sigma_1^b} APC_1$$

and the following equivalencies:

- $S_2^1 \neq BT$ iff S_2^1 does not prove $dWPHP(CV)$;
- $S_2^1 \preceq_{\Sigma_1^b} BT$ iff S_2^1 does not prove $dWPHP_1(CV, CV)$.

Furthermore, within PV , we can always find a p-time function g without parameters such that $S_2^1(PV)$ proves $dWPHP(g) \rightarrow dWPHP(f)$ for all p-time f . Namely g is the truth-table function. However it is not known whether this is true for $S_2^1(PV_1)$.

1.2 Computational Complexity Preliminaries

Now the short-awaited connection to computational complexity! Generally witnessing theorems say that some theory T proving some $\forall a \exists y A(x, y)$ with $A \in \Gamma$ implies the existence of a witness $f \in \mathcal{C}$. A witness satisfies

$$\forall x A(x, f(x))$$

Theorem 1.3. Assuming that BT is $\forall \Sigma_1^b$ -conservative over S_2^1 , then any formula from the class $dWPHP_1(PV, PV)$ can be witnessed by a p-time function.

Proof. This proof sketch is simple: choose polynomially random values of $y < 2a$. It is exponentially unlikely that none of these satisfy PV , hence we have a randomized p-time algorithm. Then with universal derandomization (assuming it is constructable in S_2^1 , which might be unlikely) we have just a p-time witnessing function. \square

Similarly to the previous connection, let us try to construct a witnessing function for $dWPHP(PV)$. It is Σ_2^b , so we require theory S_2^1 or PV_1 . The difference in these choices is meaningful: $PV_1 \neq S_2^1(PV)$ unless $NP \subset P/poly$. It turns out that if we choose theory S_2^1 , we can construct witnesses from $FP^{NP}[wit, O(\log n)]$.

There are more connections to witnessing theorems:

Theorem 1.4 (Interactive class witnessing theorem). Suppose we are given formulas of the form

$$\forall x \exists y (|y| \leq |x|^c) \forall z (|z| \leq |x|^d) A(x, y, z).$$

Then there exists a construction of a witnessing function $f(x)$ such that

$$\forall z (|z| \leq n^d) A(x, f(x), z)$$

Remark 1.3. *We care about this syntax of statements since upon appropriate choice of A the formula is $\forall\Sigma_3^b$ or $\forall\Sigma_2^b$.*

Remark 1.4. *The particular complexity class is, PCG claims, "S-T computations", which may be a dumbed down name for dumbed down readers, or maybe that's the official name, but it doesn't sound so official. It involves student S and ultimate evolved Merlin (infinitely powerful) teacher T . S sends candidates*