The strength of sharply bounded induction

Emil Jeřábek*
Mathematical Institute of AS CR
Žitná 25
115 67 Praha 1
Czech Republic
jerabek@math.cas.cz

October 24, 2006

Abstract

We prove that the sharply bounded arithmetic T^0_2 in a language containing the function symbol $\lfloor \frac{x}{2^y} \rfloor$ (often denoted by MSP) is equivalent to PV_1.

1 Introduction

The most commonly studied theories of first-order bounded arithmetic are the theories S^i_2 and T^i_2 introduced by Buss [1], which are respectively axiomatized by the schema of polynomial induction and ordinary induction for Σ^b_i-formulas over a weak base theory. Usually these theories are considered only for $i \geq 1$: while S^1_2 and its extension T^1_2 are well-behaved and fit nicely in the hierarchy, the sharply bounded arithmetics S^0_2 and T^0_2 are avoided, because they seem to lack sufficient bootstrapping power. The theory T^0_2 is typically substituted with its extension PV_1, which includes function symbols for all polynomial-time computable functions. In fact, Takeuti [14] has shown that S^0_2 is too weak to prove the existence of predecessors, which means it does not even contain Robinson's Q. Similar independence results were obtained for other variants of S^0_2 and R^0_2 by Johannsen [8, 9, 10].

We will show that the case of T^0_2 is quite different: if we extend Buss' language by the function $\lfloor \frac{x}{2^y} \rfloor$ (also known as $MSP(x, y)$, for "most significant part"), T^0_2 can Σ^b_1-define all polynomial-time functions, and PV_1 is a conservative extension of T^0_2. Thus the standard treatment of T^0_2 as an exception in Buss' hierarchy is not necessary.

Although we do not resolve the status of the original Buss' T^0_2, we believe that the inclusion of $\lfloor \frac{x}{2^y} \rfloor$ in the basic language is justified. On one hand, $\lfloor \frac{x}{2^y} \rfloor$ is a simple AC^0-function, thus it is nowhere near full polynomial time as the usual language of PV_1. On the other hand, $\lfloor \frac{x}{2^y} \rfloor$ is routinely used in the basic language of other weak theories, like R^1_2 [15]. Even the

*The research was done while the author was visiting the Department of Computer Science of the University of Toronto. Supported by NSERC Discovery grant, grant IAA1019401 of GA AV ČR, and grant 1M0545 of MŠMT ČR.

language of S^1_2 is often extended by function symbols which allow sequence coding in Σ^b_0; the equivalence of Σ^b_1 and strict Σ^b_1-formulas relies on this. We should point out that $\lfloor\frac{x}{2^y}\rfloor$ has a Σ^b_1-definition which is provably total and reasonably well-behaved in Buss' T^0_2; we only need to include the function in the language so that it can be used freely in induction axioms.

The paper is organized as follows. Section 2 provides background in bounded arithmetic, including our working definition of T^0_2 and PV_1. In section 3 we prove a few auxiliary lemmas about T^0_2. The main argument is in section 4, where we show how to define PV-functions in T^0_2, and we prove that PV_1 is a conservative extension of T^0_2. Section 5 contains concluding remarks. In the appendix we present a simplified axiom system for *IOpen*, which is used in section 3.

2 Preliminaries

As we already indicated in the introduction, we will work with the first-order language $L = \langle 0, S, +, \cdot, \leq, \#, |x|, \lfloor\frac{x}{2^y}\rfloor\rangle$. The intended meaning of the symbols is $|x| = \lceil\log_2(x+1)\rceil$ and $x\#y = 2^{|x||y|}$, the rest are the usual arithmetical operations on nonnegative integers. *Bounded quantifiers* are defined by

$$\begin{aligned}\exists x \leq t\, \varphi &\Leftrightarrow \exists x\, (x \leq t \wedge \varphi),\\ \forall x \leq t\, \varphi &\Leftrightarrow \forall x\, (x \leq t \to \varphi),\end{aligned}$$

where t is a term with no occurrence of x. A bounded quantifier is *sharply bounded*, if its bounding term t is of the form $|s|$. A formula φ is (sharply) bounded if all quantifiers in φ are (sharply) bounded. The set of all sharply bounded formulas is denoted by Σ^b_0.

The original T^0_2 is axiomatized by Σ^b_0-induction over a set of 32 open axioms called *BASIC*. We need to adjust *BASIC* to our modified language; we take this opportunity to considerably simplify the list of axioms.

Definition 2.1 T^0_2 is a theory in the language L, axiomatized by the induction schema (*IND*)

$$\varphi(0) \wedge \forall x\, (\varphi(x) \to \varphi(Sx)) \to \forall x\, \varphi(x)$$

for Σ^b_0-formulas φ, and the following formulas:

$$\begin{array}{ll}
x + 0 = x & x + Sy = S(x+y)\\
x \cdot 0 = 0 & x \cdot Sy = x \cdot y + x\\
0 \leq x & Sy \leq x \leftrightarrow y < x\\
\lfloor\frac{x}{2^0}\rfloor = x & \lfloor\frac{x}{2^y}\rfloor = 2\lfloor\frac{x}{2^{Sy}}\rfloor \vee \lfloor\frac{x}{2^y}\rfloor = S(2\lfloor\frac{x}{2^{Sy}}\rfloor)\\
|0| = 0 & x \neq 0 \to |x| = S|\lfloor\frac{x}{2}\rfloor|\\
0 \# 1 = 1 & x \neq 0 \to x \# 1 = 2(\lfloor\frac{x}{2}\rfloor \# 1) \wedge (y \# x) = (y \# 1)(y \# \lfloor\frac{x}{2}\rfloor)
\end{array}$$

where $x < y$, $\lfloor\frac{x}{2}\rfloor$, 1, 2 are abbreviations for $x \leq y \wedge x \neq y$, $\lfloor\frac{x}{2^1}\rfloor$, $S0$, $SS0$, respectively.

We remark that the dozen axioms in definition 2.1 are *not* a drop-in replacement for Buss' *BASIC*: in particular, they are most likely insufficient to bootstrap theories not based on the induction schema, such as S^1_2 or R^1_2.

PV is an equational theory intended to formalize polynomial-time reasoning, defined by Cook [5]. Its language contains several basic functions, and it is closed under composition and limited recursion on notation. It is based on an earlier result of Cobham [4], which implies that PV-functions represent the class of all polynomial-time computable functions. PV includes defining equations for its function symbols, and it has a form of induction as a derivation rule.

The theory PV_1, also denoted as QPV, $T^0_2(\square^p_1)$, and $\forall\Sigma^b_1(S^1_2)$ [12, 2, 6, 3], is a first-order variant of PV. The exact definition of the language and set of axioms of PV_1 varies in the literature, and details are often let unspecified; for definiteness, we fix the theory as follows.

Definition 2.2 The language L_{PV} contains L, and is closed under two formation rules:

- for any L_{PV}-term t whose free variables are among $x_1, \dots, x_n$, there is a function symbol $f_t(\vec{x})$,
- for any L_{PV}-function symbols $g(\vec{x})$, $h_0(\vec{x}, y, z)$, $h_1(\vec{x}, y, z)$, $b(\vec{x}, y)$, there is a function symbol $f_{g,h_0,h_1,b}(\vec{x}, y)$.

PV_1 is the theory in the language L_{PV} which consists of the axioms of T^0_2, and the following additional axioms:

- for every function symbol f_t,
$$f_t(\vec{x}) = t,$$
- for every function symbol $f_{g,h_0,h_1,b}$,
$$\begin{gathered} f_{g,h_0,h_1,b}(\vec{x}, y) \leq b(\vec{x}, y), \\ g(\vec{x}) \leq b(\vec{x}, 0) \to f_{g,h_0,h_1,b}(\vec{x}, 0) = g(\vec{x}), \\ y \neq 0 \land h_0(\vec{x}, y, f_{g,h_0,h_1,b}(\vec{x}, y)) \leq b(\vec{x}, 2y) \to f_{g,h_0,h_1,b}(\vec{x}, 2y) = h_0(\vec{x}, y, f_{g,h_0,h_1,b}(\vec{x}, y)), \\ h_1(\vec{x}, y, f_{g,h_0,h_1,b}(\vec{x}, y)) \leq b(\vec{x}, S(2y)) \to f_{g,h_0,h_1,b}(\vec{x}, S(2y)) = h_1(\vec{x}, y, f_{g,h_0,h_1,b}(\vec{x}, y)), \end{gathered}$$
- polynomial induction (*PIND*) for open formulas φ:
$$\varphi(0) \land \forall x\, (\varphi(\lfloor \tfrac{x}{2} \rfloor) \to \varphi(x)) \to \forall x\, \varphi(x).$$

As we want to compare PV_1 with T^0_2, we need to identify symbols of T^0_2 with some PV-function symbols. The easiest way is to directly include L in the basic language of PV, which motivates our definition of L_{PV}. Other details are more or less arbitrary.

3 Bootstrapping T^0_2

As T^0_2 does not include any of the usual basic theories used in the development of bounded arithmetic (such as S^1_2 or R^1_2), we have to "bootstrap" the theory first, i.e., show that it proves common auxiliary properties of the symbols in its language.

The task is simplified by observing that T^0_2 contains the well-known theory *IOpen* (see e.g. Shepherdson [13]). (More precisely, the usual axiomatization of *IOpen* is included in T^0_2 plus Buss' *BASIC*. In the appendix we provide an alternative axiomatization for *IOpen*, which shows that our weaker base theory is sufficient.) Thus, T^0_2 proves all the usual elementary properties of addition, multiplication, and ordering; we will concentrate on the other symbols, $\lfloor \frac{x}{2^y} \rfloor$, $|x|$, and $\#$.

Bounded sets of logarithmically small numbers can be encoded by numbers, using digits in their binary expansion. Formally, we define an elementhood predicate by the open formula

$$\begin{aligned} i \in x &\Leftrightarrow \lfloor \tfrac{x}{2^i} \rfloor \text{ is odd} \\ &\Leftrightarrow \lfloor \tfrac{x}{2^i} \rfloor = 2\lfloor \tfrac{x}{2^{i+1}} \rfloor + 1. \end{aligned}$$

Notice that the concept of even and odd numbers is well-behaved in T^0_2, as *IOpen* proves existence and uniqueness of division with remainder (cf. lemma A.4).

Lemma 3.1 *The following are provable in T^0_2.*

(*i*) $\lfloor \frac{x}{2^{i+j}} \rfloor = \lfloor \frac{\lfloor x/2^i \rfloor}{2^j} \rfloor$, *and as a special case,* $\lfloor \frac{x}{2^{i+1}} \rfloor = \lfloor \frac{\lfloor x/2^i \rfloor}{2} \rfloor$

(*ii*) $\lfloor \frac{x}{2^i} \rfloor \neq 0 \to |x| = i + |\lfloor \frac{x}{2^i} \rfloor|$

(*iii*) $i \geq |x| \to \lfloor \frac{x}{2^i} \rfloor = 0$

(*iv*) $x \leq y \to \lfloor \frac{x}{2^i} \rfloor \leq \lfloor \frac{y}{2^i} \rfloor$

(*v*) $i \in x \to i < |x|$

(*vi*) $x \neq 0 \to |x| - 1 \in x$

(*vii*) $i \in \lfloor \frac{x}{2^j} \rfloor \leftrightarrow i + j \in x$

(*viii*) $x \leq y \to |x| \leq |y|$

Proof: (*i*): we have

$$\lfloor \tfrac{x}{2^{i+1}} \rfloor = \lfloor \tfrac{\lfloor x/2^i \rfloor}{2} \rfloor$$

by uniqueness of division and the axioms for $\lfloor \frac{x}{2^i} \rfloor$, from which the result follows by induction on j.

(*ii*): by induction on i. For the induction step, we need the axiom $y \neq 0 \to |y| = 1 + |\lfloor \frac{y}{2} \rfloor|$, and

$$\lfloor \tfrac{x}{2^{i+1}} \rfloor \neq 0 \to \lfloor \tfrac{x}{2^i} \rfloor \neq 0,$$

which follows from the axioms for $\lfloor \frac{x}{2^i} \rfloor$.

(iii) is a corollary of (ii), and (v) is a reformulation of (iii).

(iv) follows by induction on i from basic properties of $\leq$, namely $u < v \to 2u + 1 < 2v$.

(vi): we have $\lfloor \frac{x}{2^0} \rfloor \neq 0$ and $\lfloor \frac{x}{2^{|x|}} \rfloor = 0$ from (iii), thus by induction there exists an $i < |x|$ such that

$$\lfloor \tfrac{x}{2^i} \rfloor \neq 0 = \lfloor \tfrac{x}{2^{i+1}} \rfloor.$$

The axiom for $\lfloor \frac{x}{2^i} \rfloor$ implies $\lfloor \frac{x}{2^i} \rfloor = 1$, thus $i \in x$. Moreover, $|x| = i + 1$ by (ii).

(vii) follows from (i).

$(viii)$: we have $\lfloor \frac{y}{2^{|y|}} \rfloor = 0$ by (iii), thus $\lfloor \frac{x}{2^{|y|}} \rfloor = 0$ by (iv). If $|x| > |y|$, then $\lfloor \frac{x}{2^{|x|-1}} \rfloor = 0$ by (i), contradicting (vi). □

Lemma 3.2 *The following are provable in* T^0_2.

(i) $x < y \leftrightarrow \exists i\, (i \in y \wedge i \notin x \wedge \forall j > i\, (j \in x \leftrightarrow j \in y))$,

(ii) $x = y \leftrightarrow \forall i\, (i \in x \leftrightarrow i \in y)$,

(iii) $x \leq y \leftrightarrow \forall i\, (\forall j > i\, (j \in x \leftrightarrow j \in y) \to (i \in x \to i \in y))$,

(iv) $y = Sx \leftrightarrow \exists i\, (\forall j > i\, (j \in x \leftrightarrow j \in y) \wedge i \notin x \wedge i \in y \wedge \forall j < i\, (j \in x \wedge j \notin y))$,

(v) $i \in x + y \leftrightarrow (i \in x \oplus i \in y \oplus \mathit{Carry}(i, x, y))$,

where $\alpha \oplus \beta = \neg(\alpha \leftrightarrow \beta)$ *is the exclusive-or connective, and* Carry *is the formula*

$$\exists j < i\, (j \in x \wedge j \in y \wedge \forall k < i\, (k > j \to k \in x \vee k \in y)).$$

Proof: (i): we begin with the left-to-right direction. Assume $x < y$. As $\lfloor \frac{x}{2^0} \rfloor \neq \lfloor \frac{y}{2^0} \rfloor$ and $\lfloor \frac{x}{2^{|y|}} \rfloor = \lfloor \frac{y}{2^{|y|}} \rfloor$ by lemma 3.1 (ii), induction implies that there exists an $i < |y|$ such that

$$\lfloor \tfrac{x}{2^i} \rfloor \neq \lfloor \tfrac{y}{2^i} \rfloor \text{ and } \lfloor \tfrac{x}{2^{i+1}} \rfloor = \lfloor \tfrac{y}{2^{i+1}} \rfloor.$$

Put $z = \lfloor \frac{x}{2^{i+1}} \rfloor$. We have $\lfloor \frac{x}{2^i} \rfloor < \lfloor \frac{y}{2^i} \rfloor$ by lemma 3.1 (iv), and both $\lfloor \frac{x}{2^i} \rfloor$ and $\lfloor \frac{y}{2^i} \rfloor$ are equal to $2z$ or $2z + 1$, thus

$$\lfloor \tfrac{x}{2^i} \rfloor = 2z \text{ and } \lfloor \tfrac{y}{2^i} \rfloor = 2z + 1.$$

In particular, $i \notin x$ and $i \in y$. For any $j > i$, we have $\lfloor \frac{x}{2^j} \rfloor = \lfloor \frac{y}{2^j} \rfloor$ by lemma 3.1 (i), thus $j \in x$ iff $j \in y$.

Right-to-left: fix i which witnesses the RHS. As $i \notin x \wedge i \in y$, we cannot have $x = y$. Assume for contradiction $y < x$. By the left-to-right implication there exists an i' such that

$$i' \in x \wedge i' \notin y \wedge \forall j > i'\, (j \in x \leftrightarrow j \in y).$$

Then either of $i < i'$, $i = i'$, $i > i'$ leads to a contradiction.

(ii): the left-to-right direction is trivial, and the right-to-left direction follows from (i). Likewise, (iii) is just a reformulation of (i).

(iv): by extensionality (i.e., (ii)), it suffices to prove the left-to-right direction. We have

$$\lfloor \tfrac{y}{2^j} \rfloor \le \lfloor \tfrac{x}{2^j} \rfloor + 1$$

by induction on j. Fix an i such that

$$\forall j > i\,(j \in x \leftrightarrow j \in y) \wedge i \notin x \wedge i \in y$$

by (i). Let $j < i$. We have $\lfloor \frac{x}{2^j} \rfloor < \lfloor \frac{y}{2^j} \rfloor$ by lemma 3.1 (i) and (iv), thus $\lfloor \frac{y}{2^j} \rfloor = \lfloor \frac{x}{2^j} \rfloor + 1$. By the same argument, $\lfloor \frac{y}{2^{j+1}} \rfloor = \lfloor \frac{x}{2^{j+1}} \rfloor + 1$. As $\lfloor \frac{x}{2^j} \rfloor \le 2\lfloor \frac{x}{2^{j+1}} \rfloor + 1 < 2\lfloor \frac{y}{2^{j+1}} \rfloor \le \lfloor \frac{y}{2^j} \rfloor$, we must have

$$\lfloor \tfrac{x}{2^j} \rfloor = 2\lfloor \tfrac{x}{2^{j+1}} \rfloor + 1 \text{ and } \lfloor \tfrac{y}{2^j} \rfloor = 2\lfloor \tfrac{y}{2^{j+1}} \rfloor,$$

thus $j \in x$ and $j \notin y$.

(v): we prove

$$\lfloor \tfrac{x+y}{2^i} \rfloor = \begin{cases} \lfloor \frac{x}{2^i} \rfloor + \lfloor \frac{y}{2^i} \rfloor + 1 & \text{if } \mathit{Carry}(i,x,y), \\ \lfloor \frac{x}{2^i} \rfloor + \lfloor \frac{y}{2^i} \rfloor & \text{otherwise} \end{cases}$$

by induction on $i \le |x+y|$. Let x_i, y_i, and c_i denote the indicators of the formulas $i \in x$, $i \in y$, and $\mathit{Carry}(i,x,y)$. The definition of Carry readily implies

$$\mathit{Carry}(i+1,x,y) \Leftrightarrow \big(\mathit{Carry}(i,x,y) \wedge (i \in x \vee i \in y)\big) \vee (i \in x \wedge i \in y),$$

thus

$$c_{i+1} = \mathit{MAJ}(c_i, x_i, y_i) = \lfloor \tfrac{c_i + x_i + y_i}{2} \rfloor.$$

Then

$$\begin{aligned} \lfloor \tfrac{x+y}{2^{i+1}} \rfloor = \lfloor \tfrac{\lfloor (x+y)/2^i \rfloor}{2} \rfloor = \lfloor \tfrac{\lfloor x/2^i \rfloor + \lfloor y/2^i \rfloor + c_i}{2} \rfloor = \lfloor \tfrac{2\lfloor x/2^{i+1} \rfloor + x_i + 2\lfloor y/2^{i+1} \rfloor + y_i + c_i}{2} \rfloor \\ = \lfloor \tfrac{x}{2^{i+1}} \rfloor + \lfloor \tfrac{y}{2^{i+1}} \rfloor + \lfloor \tfrac{x_i + y_i + c_i}{2} \rfloor = \lfloor \tfrac{x}{2^{i+1}} \rfloor + \lfloor \tfrac{y}{2^{i+1}} \rfloor + c_{i+1} \end{aligned}$$

by the induction hypothesis. □

We remark that all quantifiers in lemma 3.2 can be sharply bounded by lemma 3.1 (v).

Lemma 3.3 T^0_2 *proves:*

(i) Σ^b_0-*PIND*

(ii) $y \ne 0 \to |y(x \# 1)| = |x| + |y|$

(iii) $i \in y(x \# 1) \leftrightarrow i \ge |x| \wedge i - |x| \in y$

(iv) $x \# 0 = 1$

(v) $|x \# y| = |x| \cdot |y| + 1$

(vi) $i \in x \# y \leftrightarrow i = |x| \cdot |y|$

Proof: (*i*): assume $\forall x\,(\varphi(\lfloor \frac{x}{2} \rfloor) \to \varphi(x))$ and $\neg\varphi(a)$, where $\varphi \in \Sigma^b_0$. We have $\neg\varphi(\lfloor \frac{a}{2^i} \rfloor)$ by induction on i, thus $\neg\varphi(0)$ by taking $i = |a|$.

(*ii*) and (*iii*): assume $y \neq 0$, and prove

$$|y(x \# 1)| = |x| + |y| \wedge \forall i < |y(x \# 1)|\,(i \in y(x \# 1) \to i \geq |x|) \\ \wedge \forall i < |y|\,(i \in y \leftrightarrow i + |x| \in y(x \# 1))$$

by *PIND* on x. The induction step follows from $x \# 1 = 2(\lfloor \frac{x}{2} \rfloor \# 1)$ and lemma 3.1 (*vii*).

(*iv*): we have $x \# 1 \neq 0$ by (*ii*), thus $x \# 0 = 1$ follows from the axiom $x \# 1 = (x \# 1)(x \# 0)$.

(*v*) and (*vi*): we prove

$$|x \# y| = |x| \cdot |y| + 1 \wedge \forall i < |x \# y|\,(i \in x \# y \leftrightarrow i = |x| \cdot |y|)$$

by *PIND* on y. The base step follows from (*iv*), the induction step from (*ii*), (*iii*), and $x \# y = (x \# 1)(x \# \lfloor \frac{y}{2} \rfloor)$. □

4 PV-functions in T^0_2

In this section we prove our main result (theorem 4.8). The basic outline of the proof is straightforward—we will show that PV functions have Σ^b_1-definitions provably total in T^0_2, such that T^0_2 proves the recursion and induction axioms from definition 2.2. The main technical ingredient is a variant of a *bit-recursion* principle (lemma 4.2), which was already used for a similar purpose in second-order context by Cook [7]; the idea goes back to Buss [1].

Definition 4.1 A formula $\varphi(i, w, \ldots)$ is *safe for bit-recursion*, if $\varphi \in \Sigma^b_0$, and all occurrences of w in φ are inside a subformula of the form $t > i \wedge t \in w$, where t is a term not containing w.

Lemma 4.2 *If φ is safe for bit-recursion, T^0_2 proves*

$$\exists! w\,(|w| \leq |a| \wedge \forall i < |a|\,(i \in w \leftrightarrow \varphi(i, w))).$$

Proof: Uniqueness: assume that w and w' satisfy the conclusion. As $\varphi(i, w)$ depends only on the bits of w to the left of i, we can prove $\forall j < |a|\,(j \geq i \to (j \in w \leftrightarrow j \in w'))$ by reverse induction on $i \leq |a|$. Taking $i = 0$, we obtain $w = w'$ from extensionality.

Existence[1]: let $\psi(w)$ denote the Σ^b_0-formula

$$|w| \leq |a| \wedge \forall i < |a|\,(\forall j < |a|\,(j > i \to (j \in w \leftrightarrow \varphi(j, w))) \to (i \in w \to \varphi(i, w))).$$

We have $\psi(0)$ and $\neg\psi(a \# 1)$, thus there exists a w such that $\psi(w) \wedge \neg\psi(w + 1)$. We assume $|w + 1| \leq |a|$, the other case is left to the reader. By lemma 3.2 and the definition of ψ, there exist $i, k < |a|$ such that

$$\forall j > i\,(j \in w \leftrightarrow j \in w + 1) \wedge i \notin w \wedge i \in w + 1 \wedge \forall j < i\,(j \in w \wedge j \notin w + 1), \\ \forall j > k\,(j \in w + 1 \leftrightarrow \varphi(j, w + 1)) \wedge k \in w + 1 \wedge \neg\varphi(k, w + 1).$$

[1]Unlike T^i_2 for $i > 0$, we cannot use the Σ^b_0-maximization principle. It can be shown that $T^0_2 + \Sigma^b_0$-*MAX* $= T^0_2 + \Sigma^b_1$-*MAX* $= T^1_2$, cf. lemma 5.2.7 in [11].

Notice that

$$\forall j \geq i\,(\varphi(j,w) \leftrightarrow \varphi(j,w+1)),$$

as φ is safe for bit-recursion. Clearly, $k < i$ is impossible. If $k > i$, we get

$$\forall j > k\,(j \in w \leftrightarrow \varphi(j,w)) \wedge k \in w \wedge \neg\varphi(k,w),$$

contradicting $\psi(w)$. Thus $i = k$, and we have

$$\forall j \geq i\,(j \in w \leftrightarrow \varphi(j,w)),$$
$$\forall j < i\,(j \in w).$$

It remains to show $\forall j < i\,\varphi(j,w)$, which follows from $\psi(w)$ by reverse induction on j. □

The bit-recursion schema provides a simple method for introduction of poly-time computable functions in T^0_2 avoiding the hassle of Turing machinery. The details are fixed in the following definition; we aim to show that all PV-functions are definable by bit-recursion in T^0_2.

Definition 4.3 Let b and c be polynomials such that T^0_2 proves $c(\vec{n}) \geq b(\vec{n})$. Let $\varphi(i,w,\vec{x})$ be a formula safe for bit-recursion such that all occurrences of x_j in φ are inside a subterm of the form $x_j \# 1$, or a subformula of the form $t \in x_j$. We say that the function f with the graph

$$\begin{aligned} f(\vec{x}) = y \leftrightarrow |y| \leq b(|\vec{x}|) \wedge \exists w\,(|w| \leq c(|\vec{x}|) \wedge \\ \forall i < c(|\vec{x}|)\,(i \in w \leftrightarrow \varphi(i,w,\vec{x})) \wedge \forall i < b(|\vec{x}|)\,(i \in y \leftrightarrow i \in w)) \end{aligned}$$

is *defined by bit-recursion* from φ, b, and c.

Lemma 4.4 *Let f be defined by bit-recursion. Then T^0_2 proves $\forall \vec{x}\,\exists! y\, f(\vec{x}) = y$.*

Proof: By lemma 4.2 there exists a unique w such that $|w| \leq c(|\vec{x}|) \wedge \forall i < c(|\vec{x}|)\,(i \in w \leftrightarrow \varphi(i,w,\vec{x}))$, and given w, there exists a unique y such that $|y| \leq b(|\vec{x}|) \wedge \forall i < b(|\vec{x}|)\,(i \in y \leftrightarrow i \in w)$. □

Lemma 4.5 *T^0_2 proves that the class of functions defined by bit-recursion is closed under composition.*

Proof: We consider only unary functions for simplicity. Let $f_0(x)$ be defined by bit-recursion from $\varphi_0(i,w,x)$, $b_0(n)$, $c_0(n)$, let $f_1(y)$ be defined from $\varphi_1(i,w,y)$, $b_1(n)$, $c_1(n)$, we will show that $f(x) = f_1(f_0(x))$ is also defined by bit-recursion.

We define the witness for $f(x) = z$ as a concatenation of the witness for $f_0(x) = y$, the witness for $f_1(y) = z$, and z. (We need to duplicate z like this because we cannot express $b_1(|f_0(x)|)$ as a polynomial in $|x|$.) In detail, we put $b(n) = b_1(b_0(n))$, $c(n) = c_0(n) +$

$c_1(b_0(n)) + b(n)$, and we define $\varphi(i, w, x)$ as the formula

$$
\begin{aligned}
&\exists j < c_0(|x|)\, (i = j + c_1(b_0(|x|)) + b(|x|) \wedge \varphi_0'(j, w, x)) \\
\vee &\exists m < b_0(|x|)\, [(m = 0 \vee m + c_1(b_0(|x|)) + b(|x|) - 1 \in w) \\
&\qquad\qquad \wedge \forall k < b_0(|x|)\, (k \geq m \rightarrow k + c_1(b_0(|x|)) + b(|x|) \notin w) \\
&\qquad\qquad \wedge ((i < b_1(m) \wedge i + b(|x|) \in w) \vee \exists j < c_1(m)\, (i = j + b(|x|) \wedge \varphi_1'(j, w, x)))].
\end{aligned}
$$

(The conditions on the second and third line give the variable m the value $|f_0(x)|$.) Here $\varphi_0'(j, w, x)$ is constructed from $\varphi_0(j, w, x)$ by replacing each subformula $t \in w$ with $t + c_1(b_0(|x|)) + b(|x|) \in w$, and $\varphi_1'(j, w, x)$ is constructed from $\varphi_1(j, w, y)$ by replacing subformulas $t \in w$ with $t + b(|x|) \in w$, subformulas $t \in y$ with $t < b_0(|x|) \wedge t + c_1(b_0(|x|)) + b(|x|) \in w$, and subterms $y \# 1$ with 2^m. We can express 2^m as follows: by lemma 3.3, there is a term t such that $2^{b_0(|x|)} = t(x \# 1)$, and we put $2^m = \left\lfloor \frac{t(x\#1)}{2^{b_0(|x|)-m}} \right\rfloor$. (Subtraction here and above should be simulated by a quantifier.)

Using lemma 3.3, we can make all quantifiers in φ sharply bounded, and it is also easy to see that φ is safe for bit-recursion. The conditions on x from definition 4.3 are also satisfied: all occurrences of x are either of the form $t \in x$, or inside a subterm $x \# 1$ or $|x|$; the latter is equal to $|\lfloor \frac{x\#1}{2} \rfloor|$.

Notice that $|f(x)| \leq b_1(|f_0(x)|) \leq b_1(b_0(|x|)) = b(|x|)$ as b_1 is monotone. If w is constructed by bit-recursion using $\varphi(i, w, x)$, it is easy to see that bits $c_1(b_0(|x|)) + b(|x|)$ up to $c(|x|) - 1$ of w give a witness for $f_0(x) = y$, bits $b(|x|)$ up to $c_1(b_0(|x|)) + b(|x|) - 1$ give a witness for $f_1(y) = z$, and bits below $b(|x|)$ give z, thus φ, b, and c define f by bit-recursion. □

Lemma 4.6 *Let t be an L-term, and f a function defined by bit-recursion. Then T^0_2 proves*

$$f(\vec{x}, 0) = t(\vec{x}, 0) \wedge \forall u \leq y\, (f(\vec{x}, \lfloor \tfrac{u}{2} \rfloor) = t(\vec{x}, \lfloor \tfrac{u}{2} \rfloor) \rightarrow f(\vec{x}, u) = t(\vec{x}, u)) \rightarrow f(\vec{x}, y) = t(\vec{x}, y).$$

Proof: Assume that f is defined from $\varphi(i, w, \vec{x}, y)$, b, and c. Put $C = c(|\vec{x}|, |y|)$. Using lemma 4.2, find w such that $|w| \leq (|y| + 1)C$ and for every $j \leq |y|$ and $i < C$,

$$i + jC \in w \Leftrightarrow i < c(|\vec{x}|, |\lfloor \tfrac{y}{2^j} \rfloor|) \wedge \varphi'(i, w, \vec{x}, \lfloor \tfrac{y}{2^j} \rfloor),$$

where φ' is obtained from φ by replacing subformulas $s \in w$ with $s < c(|\vec{x}|, |\lfloor \frac{y}{2^j} \rfloor|) \wedge s + jC \in w$. Then we can prove

$$|t(\vec{x}, \lfloor \tfrac{y}{2^j} \rfloor)| \leq b(|\vec{x}|, |\lfloor \tfrac{y}{2^j} \rfloor|) \wedge \forall i < b(|\vec{x}|, |\lfloor \tfrac{y}{2^j} \rfloor|)\, (i + jC \in w \leftrightarrow i \in t(\vec{x}, \lfloor \tfrac{y}{2^j} \rfloor))$$

by reverse induction on $j \leq |y|$; the case $j = 0$ yields $f(\vec{x}, y) = t(\vec{x}, y)$. □

Lemma 4.7 *Let $b(\vec{n}, m)$ be a polynomial, and $g(\vec{x})$, $h(\vec{x}, y, z)$ functions defined by bit-recursion. There exists a function $f(\vec{x}, y)$ defined by bit-recursion such that T^0_2 proves*

$$
\begin{gathered}
|g(\vec{x})| \leq b(|\vec{x}|, 0) \rightarrow f(\vec{x}, 0) = g(\vec{x}), \\
y \neq 0 \wedge |h(\vec{x}, y, f(\vec{x}, \lfloor \tfrac{y}{2} \rfloor))| \leq b(|\vec{x}|, |y|) \rightarrow f(\vec{x}, y) = h(\vec{x}, y, f(\vec{x}, \lfloor \tfrac{y}{2} \rfloor)).
\end{gathered}
$$

Proof: Assume that g is defined from φ_0, b_0, c_0, and h is defined from φ_1, b_1, c_1. We put $c(\vec{n},m) = c_0(\vec{n}) + mc_1(\vec{n},m,b(\vec{n},m)) + (m+1)b(\vec{n},m)$. We let φ formalize the following description (we omit the details, which are similar to lemmas 4.5 and 4.6): the witness for $f(\vec{x},y) = z$ is the concatenation $w_{|y|} \frown z_{|y|} \frown w_{|y|-1} \frown z_{|y|-1} \frown \cdots \frown w_0 \frown z_0$, where z_i is $f(\vec{x}, \lfloor \frac{y}{2^i} \rfloor)$ padded to $b(|\vec{x}|,|y|)$ bits, $w_{|y|}$ is the witness for $z_{|y|} = g(\vec{x})$, and for $i < |y|$, w_i is the witness for $z_i = h(\vec{x}, \lfloor \frac{y}{2^i} \rfloor, z_{i+1})$ padded to $c_1(|\vec{x}|,|y|,b(|\vec{x}|,|y|))$ bits. Whenever $|z_i|$ would exceed $b(|\vec{x}|, |\lfloor \frac{y}{2^i} \rfloor|)$, z_i is replaced with 0.

To show the recursion identity, we argue as follows: let w be the witness for $f(\vec{x},y) = z_0$. We use lemma 4.2 to cut the last part $w_0 \frown z_0$ from w, and to shorten the remaining blocks to the appropriate length, and observe that the result is the witness for $f(\vec{x}, \lfloor \frac{y}{2} \rfloor) = z_1$. By construction of w, $z_0 = h(\vec{x}, y, z_1)$. □

Theorem 4.8 *PV_1 is an extension of T^0_2 by definitions. In particular, PV_1 is conservative over T^0_2.*

Proof: We expand T^0_2 by symbols for all functions definable by bit-recursion, we will prove that the resulting theory contains PV_1.

If t is a term made of functions defined by bit-recursion, then f_t is definable by bit-recursion by lemma 4.5.

Assume that $g(\vec{x})$, $h_0(\vec{x},y,z)$, $h_1(\vec{x},y,z)$, $b(\vec{x},y)$ are defined by bit-recursion, and let $f_{g,h_0,h_1,b}(\vec{x},y)$ be defined by limited recursion on notation as in definition 2.2. By definition 4.3, there is a polynomial p such that T^0_2 proves $|b(\vec{x},y)| \le p(|\vec{x}|,|y|)$. Using lemma 3.2, we see that the functions $\lfloor \frac{x}{2} \rfloor$ and

$$\mathrm{cond}(u,x,y) = \begin{cases} x, & \text{if } u = 0, \\ y, & \text{otherwise,} \end{cases}$$

$$\mathrm{mod2}(x) = \begin{cases} 1, & \text{if } x \text{ is odd,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\min(x,y) = \begin{cases} x, & \text{if } x \le y, \\ y, & \text{otherwise} \end{cases}$$

are bit-definable (i.e., definable by bit-recursion from a formula $\varphi(i,w,\vec{x})$ which does not involve w), thus by lemma 4.5, there is a function h defined by bit-recursion such that T^0_2 proves

$$h(\vec{x},y,z) = \min(b(\vec{x},y), \mathrm{cond}(\mathrm{mod2}(y), h_0(\vec{x}, \lfloor \tfrac{y}{2} \rfloor, z), h_1(\vec{x}, \lfloor \tfrac{y}{2} \rfloor, z))).$$

By lemma 4.7, there is a function f defined by bit-recursion such that T^0_2 proves

$$|\min(b(\vec{x},0), g(\vec{x}))| \le p(|\vec{x}|,0) \to f(\vec{x},0) = \min(b(\vec{x},0), g(\vec{x})),$$
$$y \ne 0 \wedge |h(\vec{x},y,f(\vec{x}, \lfloor \tfrac{y}{2} \rfloor))| \le p(|\vec{x}|,|y|) \to f(\vec{x},y) = h(\vec{x},y,f(\vec{x}, \lfloor \tfrac{y}{2} \rfloor)).$$

The choice of p ensures that T^0_2 proves the conditions $|\cdots| \le p(\cdots)$, and consequently T^0_2 proves that f satisfies the axioms for $f_{g,h_0,h_1,b}$ from definition 2.2.

Basic functions of L_{PV} are already included as function symbols in L, nevertheless we need to show that they are definable by bit-recursion so that we can apply composition and limited recursion on notation to these function symbols. The functions 0, S, $+$, $\lfloor \frac{x}{2^y} \rfloor$, and $\#$ are bit-definable by lemmas 3.1, 3.2, and 3.3. The function $|x|$ is trivially bit-definable, because we may use $|x| = |\lfloor \frac{x \# 1}{2} \rfloor|$ freely according to definition 4.3. Using limited recursion on notation, there is a function f defined by bit-recursion such that T^0_2 proves

$$\begin{gathered} |f(x,y)| \leq |x| + |y|, \\ f(x,0) = 0, \\ y \neq 0 \wedge |2f(x,y)| \leq |x| + |2y| \rightarrow f(x,2y) = 2f(x,y), \\ |2f(x,y) + x| \leq |x| + |2y+1| \rightarrow f(x,2y+1) = 2f(x,y) + x. \end{gathered}$$

Then we can prove $f(x,y) = xy$ by *PIND* on y (i.e., by lemma 4.6); the bound $|xy| \leq |x| + |y|$ follows from lemma 3.3 (ii).

It remains to show that T^0_2 proves *PIND* for open formulas φ of the language L_{PV}. By lemma 4.6, it suffices to show that each such φ has a characteristic function, i.e., there is a function f defined by bit-recursion such that T^0_2 proves

$$\begin{gathered} \varphi(\vec{x}) \rightarrow f(\vec{x}) = 1, \\ \neg\varphi(\vec{x}) \rightarrow f(\vec{x}) = 0. \end{gathered}$$

Characteristic functions for atomic formulas can be constructed by composition with characteristic functions of the predicates $=$ and $\leq$, which are bit-definable by lemma 3.2. Then we proceed by induction on the complexity of the formula, using composition with the functions $\mathrm{not}(x) = \mathrm{cond}(x,1,0)$, and $\mathrm{or}(x,y) = \mathrm{cond}(x,y,1)$. □

5 Remarks

Cook [7] defined a second-order arithmetical theory TV^0, and noted that PV_1 is *RSUV*-isomorphic to TV^0. This can be used to give an alternative, indirect proof of our main result: instead of constructing *PV*-functions in T^0_2, it suffices to show that T^0_2 is *RSUV*-isomorphic to TV^0. A key element in the proof is again lemma 4.2, which implies that T^0_2 proves the translation of the Σ^B_0-comprehension axioms of V^0.

Our development of T^0_2 can be carried out in Buss' language to some extent. Let T^0_2 denote the original Buss' theory for the remainder of this section. We can define the graph of exponentiation by the open formula

$$y = 2^x \Leftrightarrow |y| = x + 1 \wedge 2y = 1 \# y.$$

T^0_2 proves that 2^x is a partial function whose domain is a cut closed under multiplication, and other elementary properties of 2^x, such as $2^{x+y} = 2^x 2^y$, and $x \# y = 2^{|x||y|}$. As T^0_2 extends *IOpen*, integer division is a well-defined function, and we may introduce $\lfloor \frac{x}{2^y} \rfloor$ by

$$\lfloor \tfrac{x}{2^y} \rfloor = \begin{cases} 0, & y \geq |x|, \\ z, & \exists u \leq x \, (u = 2^y \wedge zu \leq x < (z+1)u). \end{cases}$$

We can show that T^0_2 proves lemma 3.1, most of lemma 3.3, and a slightly weaker version of lemma 3.2, among others.

Nevertheless, it is unclear how could T^0_2 prove stronger principles such as Σ^b_0-*PIND*, or lemma 4.2. As for the latter, we do not even know whether T^0_2 proves its simple instance

$$\forall x, y\, \exists w\, \forall i\, (i \in w \leftrightarrow (i \in x \oplus i \in y)).$$

On the other hand, the methods of Takeuti and Johannsen seem inadequate to show independence results for T^0_2, or even for $L^0_2 + \mathit{IOpen}$ (here L^i_2 is $\mathit{BASIC} + \Sigma^b_i$-*LIND*). The strength of Buss' T^0_2 thus remains an open question.

6 Acknowledgement

I am indebted to Jan Johannsen, Steve Cook, and Jan Krajíček for clarifying discussions, and comments on an earlier draft of this paper. I also wish to thank the anonymous referees for useful suggestions.

References

[1] Samuel R. Buss, *Bounded arithmetic*, Bibliopolis, Naples, 1986.

[2] ———, *Relating the bounded arithmetic and polynomial time hierarchies*, Annals of Pure and Applied Logic 75 (1995), no. 1–2, pp. 67–77.

[3] ———, *First-order proof theory of arithmetic*, in: Handbook of Proof Theory (S. R. Buss, ed.), Studies in Logic and the Foundations of Mathematics vol. 137, Elsevier, Amsterdam, 1998, pp. 79–147.

[4] Alan Cobham, *The intrinsic computational difficulty of functions*, in: Proceedings of the 2nd International Congress of Logic, Methodology and Philosophy of Science (Y. Bar-Hillel, ed.), North–Holland, 1965, pp. 24–30.

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

[6] ———, *Relating the provable collapse of* $\mathbf{P}$ *to* $\mathbf{NC}^1$ *and the power of logical theories*, in: Proof Complexity and Feasible Arithmetics (P. Beame and S. R. Buss, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science vol. 39, American Mathematical Society, 1998, pp. 73–92.

[7] ———, *Theories for complexity classes and their propositional translations*, in: Complexity of computations and proofs (J. Krajíček, ed.), Quaderni di Matematica vol. 13, Seconda Universita di Napoli, 2004, pp. 175–227.

[8] Jan Johannsen, *On the weakness of sharply bounded polynomial induction*, in: Computational Logic and Proof Theory, Proceedings of Kurt Gödel Colloquium '93 (G. Gottlob, A. Leitsch, and D. Mundici, eds.), Lecture Notes in Computer Science vol. 713, Springer, 1993, pp. 223–230.

[9] ———, *A note on sharply bounded arithmetic*, Archive for Mathematical Logic 33 (1994), no. 2, pp. 159–165.

[10] ———, *A model-theoretic property of sharply bounded formulae, with some applications*, Mathematical Logic Quarterly 44 (1998), no. 2, pp. 205–215.

[11] Jan Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications vol. 60, Cambridge University Press, 1995.

[12] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti, *Bounded arithmetic and the polynomial hierarchy*, Annals of Pure and Applied Logic 52 (1991), no. 1–2, pp. 143–153.

[13] John C. Shepherdson, *A nonstandard model for a free variable fragment of number theory*, Bulletin de l'Académie Polonaise des Sciences 12 (1964), no. 2, pp. 79–86.

[14] Gaisi Takeuti, *Sharply bounded arithmetic and the function $a \dot{-} 1$*, in: Logic and Computation, Proceedings of a Workshop held at Carnegie Mellon University, June 30–July 2, 1987 (W. Sieg, ed.), Contemporary Mathematics vol. 106, American Mathematical Society, 1990, pp. 281–288.

[15] ———, *RSUV isomorphisms*, in: Arithmetic, Proof Theory, and Computational Complexity (P. Clote and J. Krajíček, eds.), Oxford Logic Guides vol. 23, Oxford University Press, 1993, pp. 364–386.

A On *IOpen*

The theory *IOpen* is usually axiomatized by the schema of open induction over the theory of discretely ordered commutative semirings (or the slightly stronger theory of nonnegative parts of discretely ordered commutative rings). Here we present a simplified axiom system for *IOpen*.

Definition A.1 *IOpen'* is the theory in the language $\langle 0, S, +, \cdot, \leq \rangle$, axiomatized by

(A0) $$x + 0 = x$$

(AS) $$x + Sy = S(x + y)$$

(M0) $$x \cdot 0 = 0$$

(MS) $$x \cdot Sy = x \cdot y + x$$

(O0) $$0 \leq x$$

(OS) $$Sy \leq x \leftrightarrow y < x$$

and the induction schema for open formulas, where $x < y$ stands for $x \leq y \wedge x \neq y$.

We are going to show $IOpen' = IOpen$. The only non-obvious part is to derive the missing successor axioms of Robinson's arithmetic.

Lemma A.2 *$IOpen'$ proves:*

(i) $Sx \not\leq x$

(ii) $Sx \neq 0$

(iii) $x \leq y \to x \leq z \vee z \leq y$

(iv) $x \leq x$

(v) $x \leq y \vee y \leq x$

(vi) $Sx = Sy \to x = y$

(vii) $x \neq 0 \to \exists y\, x = Sy$

Proof: (i) follows from OS, as $x = x$.

(ii): assuming $Sx = 0$, we obtain $Sx \leq x$ from O0, contradicting (i).

(iii): by induction on x. For the base step, we have $0 \leq z$ by O0. The induction step: assume $Sx \leq y$. Then $x \leq y$ by OS, thus $x \leq z$ or $z \leq y$ by the induction hypothesis. In the latter case, we are done. In the former case, we have $Sx \leq z$ or $x = z$ by OS. Finally, $x = z$ implies $z \leq y$, as $x \leq y$.

(iv): by induction on x. $0 \leq 0$ follows from O0. Assume $x \leq x$. We have $x \leq Sx$ by (iii) and (i). If $x = Sx$, we have $Sx \leq Sx$ from $x \leq x$; otherwise $x < Sx$, thus $Sx \leq Sx$ by OS.

(v) follows from (iv) and (iii).

(vi): assume for contradiction $x \neq y$. By (v), we have $x < y$ or $y < x$. If $x < y$, we obtain $Sy = Sx \leq y$ from OS, contradicting (i). The other case is symmetric.

(vii): assume $x \neq 0$ and $\forall y\, x \neq Sy$. By induction on y, we have $\forall y\, y \neq x$, in particular $x \neq x$, which is a contradiction. $\square$

The rest is straightforward:

Lemma A.3 *$IOpen'$ proves the following formulas, and consequently, $IOpen' = IOpen$.*

$(x+y)+v = x+(y+v)$	$x+v = y+v \to x = y$
$0+v = v$	$x+v \leq x \to v = 0$
$Sx+v = S(x+v)$	$x \leq y \leftrightarrow \exists z\, x+z = y$
$x+v = v+x$	$x \leq y \leq z \to x \leq z$
$x(y+v) = xy+xv$	$x \leq y \leq x \to x = y$
$(xy)v = x(yv)$	$x \leq y \leftrightarrow x+z \leq y+z$
$0v = v$	$x \leq y \to xz \leq yz$
$Sx \cdot v = xv+v$	$z \neq 0 \wedge xz \leq yz \to x \leq y$
$xv = vx$	

Proof: By induction on v and/or using the formulas proved earlier, left column first. The only exception is the formula involving an existential quantifier, which can be derived as follows.

Left-to-right: we have $x + 0 \leq y$, and $x + Sy \nleq y$, thus by induction for the formula $x + v \leq y$, there exists a z such that $x + z \leq y$ and $x + Sz \nleq y$. Then $x + z = y$ by OS.

Right-to-left: assume for contradiction $x \nleq y$. Then $y < x$, thus $Sy \leq x$. By the first part, there exists a w such that $y + Sw = x$, thus $y = y + (z + Sw)$, which implies $S(z + w) = 0$, a contradiction. □

The following lemma, which was used in sections 3 and 4, is quite standard. We include its proof for the sake of completeness.

Lemma A.4 *IOpen proves*

(i) $x \leq y \rightarrow \exists! z\,(x + z = y)$

(ii) $y \neq 0 \rightarrow \exists! u, v\,(v < y \wedge x = uy + v)$

Proof: (i) was derived in lemma A.3, we will prove (ii).

Existence: we have $0y \leq x$ and $Sx \cdot y \nleq x$ as $x < Sx \leq Sx \cdot y$. Thus by induction, there exists a u such that $uy \leq x$ and $Su \cdot y \nleq x$. By (i), there is a v such that $x = uy + v$. As $x < Su \cdot y = uy + y$, we must have $v < y$.

Uniqueness: assume $x = uy + v = u'y + v'$, where $v, v' < y$. If $u < u'$, we get $x < uy + y = Su \cdot y \leq u'y \leq x$, a contradiction. By symmetry, $u' < u$ is also impossible, thus $u = u'$. Then $uy + v = uy + v'$, thus $v = v'$. □