*Boletín de Matemáticas*
Nueva Serie, Volumen IV (1997), pp. 73–84

# RECURSION IN SECOND ORDER BOUNDED ARITHMETIC

RODRIGO DE CASTRO(*)

**Resumen.** Se muestra que algunos esquemas recursivos pueden ser ejecutados en las teorías $U_2^i$ $(i \geq 1)$ de aritmética acotada de segundo orden introducidas por S. Buss. En particular, se demuestra que la clase de las funciones $\Sigma_1^{1,b}$-definibles en $U_2^i$ es cerrada bajo recursión acotada, o, equivalentemente, que $U_2^i$ puede $\Sigma_1^{1,b}$-definir $\mathcal{E}^2$, la segunda clase Grzegorczyk.

*Abstract. It is shown that some recursion schemes can be carried out in the second order theories of Bounded Arithmetic $U_2^i$ $(i \geq 1)$ introduced by S. Buss in [2]. In particular, we prove that the class of $\Sigma_1^{1,b}$-definable functions in $U_2^i$ is closed under bounded recursion, or, equivalently, that $U_2^i$ can $\Sigma_1^{1,b}$-define the functions in the second class of Grzegorczyk, $\mathcal{E}^2$.*

*Keywords. Bounded Arithmetic, second order theories, bounded recursion, Grzegorczyk classes.*

## 1. Preliminaries

The first order language of Bounded Arithmetic introduced by S. BUSS in [2] contains all the usual logical symbols $\wedge$, $\vee$, $\neg$, $\supset$, $=$, $\exists$, $\forall$, parentheses, the non logical function symbols $S$, $0$, $+$, $\cdot$, $|x|$, $\lfloor \frac{1}{2}x \rfloor$, and $\sharp$ and the non logical predicate symbol $\leq$. The intended meaning of the non logical symbols is as follows: $S$, $0$, $+$, $\cdot$ and $\leq$ are the successor function, the zero constant, addition, multiplication, and the less-than-or-equal-to relation. $|x|$ denotes the length of the binary representation of $x$, $\lfloor \frac{1}{2}x \rfloor$ denotes the greatest integer less than or equal to $x/2$, and $x \sharp y$ is defined to be $2^{|x|\cdot|y|}$.

(*)Texto recibido 26/9/98, revisado 21/5/99. Rodrigo De Castro, Departamento de Matemáticas y Estadística, Universidad Nacional, Bogotá, Colombia. e-mail: rdcastro@matematicas.unal.edu.co The author received partial support from CINDEC (Universidad Nacional de Colombia), project # 803072.

BUSS also considered second order theories by allowing second order predicate and function variables; quantification over these variables is restricted. The function symbols are intended to range over the set of functions having polynomial growth rate.

**Definition 1.1.** *The language of second order Bounded Arithmetic consists of all symbols of first order Bounded Arithmetic mentioned above plus the following second order variables and quantifiers:*

(1) *Free and bound second order variables for predicates. For all $i$, $j \in \mathbb{N}$, $\alpha_i^j$ is a free $j$-ary second order predicate symbol and $\phi_i^j$ is a bound $j$-ary second order predicate symbol. We shall use $\alpha$, $\beta$, $\gamma, \ldots$ and $\phi$, $\chi$, $\psi, \ldots$ as metavariables for free and bound predicate variables, respectively.*
(2) *Free and bound second order variables for functions with polynomial growth rate. For every term $t$ of first order Bounded Arithmetic and for all $i$, $j \in \mathbb{N}$, $\zeta_{i,j}^t$ is a free second order $j$-ary function variable and $\lambda_{i,j}^t$ is a bound second order $j$-ary function variable. We use $\zeta^t, \eta^t, \theta^t, \ldots$ and $\lambda^t, \mu^t, \nu^t, \ldots$ as metavariables for free and bound second order function variables, respectively (in informal arguments, for both). When $t$ is understood or immaterial, it is omitted. These symbols range over functions $f$ such that $f$ is bounded by $t$; i.e., for all $\vec{x} \in \mathbb{N}^j$, $f(\vec{x}) \leq t(\vec{x})$.*
(3) *Second order quantifiers are of the form $(\forall\phi)$, $(\exists\phi)$, $(\forall\lambda^t)$ and $(\exists\lambda^t)$.*

**Definition 1.2.** *A first order formula is one having no second order quantifiers. Second order free variables may appear in a first order formula. A second order formula is bounded if it contains no unbounded, first order quantifiers.*

Second order formulae can be classified in a hierarchy of sets, $\Sigma_i^{1,b}$, $\Pi_i^{1,b}$ as follows: $\Sigma_0^{1,b} = \Pi_0^{1,b} = \Delta_0^{1,b}$ is the set of formulae which contain no second order quantifiers and no unbounded quantifiers (*i.e.*, the set of bounded, first order formulae). $\Sigma_i^{1,b}$ and $\Pi_i^{1,b}$ are defined by counting alternations of second order quantifiers ignoring first order bounded quantifiers, in a manner similar to the formation of the well known first order arithmetical hierarchy.

**Definition 1.3.** *Let $\Phi$ be a set of formulae. The $\Phi$-PIND axioms are*

$$A(0) \ \wedge \ (\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor) \supset A(x)) \ \supset \ (\forall x)\, A(x),$$

*where $A$ is any formula in $\Phi$.*

*The $\Phi$-comprehension axioms, $\Phi$-CA, are given by the axiom scheme*

$$(\forall \vec{z})(\forall \vec{\phi})(\forall \vec{\lambda^s})(\exists \chi)(\forall \vec{y})[\chi(\vec{y}) \ \leftrightarrow \ A(\vec{y}, \vec{z}, \vec{\phi}, \vec{\lambda^s})],$$

*where A is in* $\Phi$.

*The* $\Phi$*-function-comprehension axioms,* $\Phi$*-FCA, are given by the following axiom scheme*

$$(\forall \vec{z})(\forall \vec{\phi})(\forall \vec{\eta^s})(\exists \lambda^t)(\forall \vec{y})[A(\lambda^t(\vec{y}), \vec{y}, \vec{z}, \vec{\phi}, \vec{\eta^s}) \leftrightarrow (\exists x \leq t)\, A(x, \vec{y}, \vec{z}, \vec{\phi}, \vec{\eta^s})],$$

*where A is in* $\Phi$ *and t is any term of first order Bounded Arithmetic.*

The comprehension axioms can be presented as inference rules ($\Phi$-comprehension and $\Phi$-function-comprehension rules) and can be included in a natural deduction system for second order Bounded Arithmetic (see [2]).

**Definition 1.4.** *A hierarchy of the second order formulae,* $\Sigma_i^b(\alpha, \zeta)$ *and* $\Pi_i^b(\alpha, \zeta)$*, containing no second order quantifiers, can be defined in a completely analogous way to the definition of* $\Sigma_i^b$ *and* $\Pi_i^b$. *The only difference is that free second order variables may appear without restriction in the formulae. The sets* $\Sigma_i^b(\alpha)$ *and* $\Pi_i^b(\alpha)$ *contain those formulae of* $\Sigma_i^b(\alpha, \zeta)$ *and* $\Pi_i^b(\alpha, \zeta)$*, respectively, which have no second order function variables.*

**Definition 1.5.** $S_2^i(\alpha, \zeta)$ *is the second order theory with second order function and predicate variables and the following axioms:*

(1) *BASIC axioms (that is, a finite set of true open formulae of arithmetic which are sufficient to define the simple properties relating the function and predicate symbols of Bounded Arithmetic; for specifics see [2]).*
(2) *For each function variable* $\zeta_i^t$, *the axiom* $(\forall \vec{x})(\zeta_i^t(\vec{x}) \leq t(\vec{x}))$.
(3) *The* $\Sigma_i^b(\alpha, \zeta)$*-PIND axioms.*

$S_2(\alpha, \zeta)$ *is the theory* $\bigcup_i S_2^i(\alpha, \zeta)$.

**Definition 1.6.** $U_2^i$ *is the second order theory of Bounded Arithmetic which has second order predicate variables and function variables and which has the following axioms:*

(1) *All axioms of* $S_2(\alpha, \zeta)$.
(2) $\Delta_0^{1,b}$*-comprehension axioms, (*$\Delta_0^{1,b}$*-CA and* $\Delta_0^{1,b} - FCA$*).*
(3) $\Sigma_i^{1,b}$*-PIND axioms.*

**Definition 1.7.** *Let* $A(\vec{x}, y)$ *be a formula whose only free first order variables are* $\vec{x}$ *and* $y$. *We say that A defines a function in* $U_2^i$ *if there is a term* $t(\vec{x})$ *of first order Bounded Arithmetic such that*

$$(a) \quad U_2^i \vdash (\forall \vec{x})(\exists y \leq t(\vec{x}))\, A(\vec{x}, y).$$

$$(b) \quad U_2^i \vdash (\forall \vec{x})(\forall y)(\forall z)(A(\vec{x}, y) \wedge A(\vec{x}, z) \supset y = z).$$

*If A is a* $\Sigma_i^{1,b}$*-formula, we say that A* $\Sigma_i^{1,b}$*-defines a function in* $U_2^i$.

*If $f$ is a number theoretic function, we say that $f$ is definable in $U_2^i$ by the formula $A$ if $(a), (b)$ above hold and $\mathbb{N} \models \forall \vec{x}\ A(\vec{x}, f(\vec{x}))$. If $A$ is a $\Sigma_i^{1,b}$-formula, we say that $f$ is $\Sigma_i^{1,b}$-definable in $U_2^i$.*

One of the main results on second order Bounded Arithmetic proved by Buss in [2] relates the $\Sigma_1^{1,b}$-definable functions in $U_2^1$ with the complexity class PSPACE. In computer science literature PSPACE and the remaining central complexity classes, are taken to be sets of predicates. In the current context, however, they are defined as sets of functions.

By a *polynomial* we always mean a polynomial with nonnegative integer coefficients, and a function $f : \mathbb{N}^k \longrightarrow \mathbb{N}$ is said to have *polynomial growth rate* iff there is a polynomial $p(x_1, \ldots, x_k)$ such that for all $\vec{x}$, $|f(\vec{x})| \leq p(|\vec{x}|)$ holds.

**Definition 1.8.** *PSPACE is the set of number-theoretical functions $f$ which can be computed by a Turing machine $M_f$ such that there is a polynomial $p(\vec{n})$ so that the total number of tape squares used by $M_f$ on input $\vec{x}$ is always less than $p(|\vec{x}|)$.*

According to this definition a function $f$ in PSPACE has polynomial growth rate because the output $f(\vec{x})$ occupies less than $p(|\vec{x}|)$ computation squares on input $\vec{x}$. Note also that if $f$ is $\Sigma_i^{1,b}$-definable in $U_2^i$, then $f$ has polynomial growth as well because $f(\vec{x}) \leq t(\vec{x})$ (where $t$ is the term of definition 1.7) and any term of first order Bounded Arithmetic is bounded above by $2^{p(|\vec{x}|)}$ for some appropriate polynomial $p$.

**Theorem 1.9.** *(S. Buss). A number-theoretical function $f$ is in PSPACE if and only if $f$ is $\Sigma_1^{1,b}$-definable in $U_2^1$.*

*Proof.* See [2].

## 2. Bounded recursion

**Definition 2.1.** *Let $b$, $g$ and $h$ be number-theoretic functions. The function $f$ given by*

$$f(0, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\},$$
$$f(x+1, \vec{y}) = \min\{h(x+1, \vec{y}, f(x, \vec{y})), b(x+1, \vec{y})\},$$

*is said to be obtained from $b$, $g$ and $h$ by bounded recursion.*

The following theorem shows that the class of $\Sigma_1^{1,b}$-definable functions in $U_2^1$ is closed under bounded recursion. We will infer that $U_2^1$ can $\Sigma_1^{1,b}$-define an important class of primitive recursive functions, namely, the second class of Grzegorczyk, $\mathcal{E}^2$.

**Theorem 2.2.** *If the functions $b$, $g$ and $h$ are $\Sigma_1^{1,b}$-definable in $U_2^1$, then the function $f$ obtained by bounded recursion from $b$, $g$ and $h$, that is, the function $f$ such that*

$$\forall\vec{y}\ f(0,\vec{y}) = \min\{g(\vec{y}),\ b(0,\vec{y})\},$$
$$(\forall x)(\forall\vec{y})\ f(x+1,\vec{y}) = \min\{h(x+1,\vec{y}, f(x,\vec{y})),\ b(x+1,\vec{y})\}$$

*is $\Sigma_1^{1,b}$-definable in $U_2^1$.*

*Proof.* We will present two different proofs of this result. The first, a relatively simple one, uses Buss Main Theorem (Theorem 1.9), the second one is a direct proof in the theory $U_2^1$.

*First Proof.* The idea of this proof is, of course, to show that on input $x$, $\vec{y}$ the above recursion can be performed by a Turing machine within space $p(|x|, |\vec{y}|)$ where $p$ is a polynomial. On input $x$ (we treat $\vec{y}$ as parameters) this recursion scheme has depth $x+1$; however, we will see that it can be implemented in polynomial space *in the length of* $x$ by freeing (*i.e.* writing over) previously used space.

By Theorem 1.9, there are PSPACE Turing machines $M_g$, $M_b$ and $M_h$ computing $g$, $b$ and $h$ (respectively) with working space delimited by polynomials $q_g$, $q_b$ and $q_h$ (respectively).

On input $x$, $\vec{y}$, assume that $f(u,\vec{y})$, with $u < x$, has been computed and its value entered on an auxiliary output tape $T$. This value occupies less than $q_b(|u|, |\vec{y}|)$ squares because of the bound $b$ imposed on $f$. In order to compute $f(u+1,\vec{y})$ we run machines $M_b$ and $M_h$ in a tape (or tapes) other than $T$. Less than

$$q_h(|u+1|, |\vec{y}|, q_b(|u|, |\vec{y}|)) + 2q_b(|u+1|, |\vec{y}|)$$

squares are needed. We then place this output on the special tape $T$ (writing over the existing data). To continue this iteration up to $x$ we only need the information stored on tape $T$, all of the remaining squares on all of the additional tapes can be reused for the next step calculation.

It can be seen in this way that on input $x$, $\vec{y}$, the computation of $f(x,\vec{y})$ requires *at most*

$$|x| + |\vec{y}| + q_g(|\vec{y}|) + q_h(|x|, |\vec{y}|, q_b(|x|, |\vec{y}|)) + 2q_b(|x|, |\vec{y}|)$$

computation space.

*Second Proof.* Suppose that $b$ is $\Sigma_1^{1,b}$-definable by the formula $B(x,\vec{y},z)$. Then there is a term, $t(x,\vec{y})$, of first order Bounded Arithmetic (involving only the original function symbols), such that

$$U_2^1 \vdash (\forall x)(\forall\vec{y})(\exists! z \le t(x,\vec{y}))\, B(x,\vec{y},z),$$
$$U_2^1 \vdash (\forall x)(\forall\vec{y})\, b(x,\vec{y}) \le t(x,\vec{y}).$$

The defining axiom for $b$ is $b(x, \vec{y}) = z \leftrightarrow B(x, \vec{y}, z)$. Let $x$, $\vec{y}$ be fixed (but arbitrary) and let $C(\lambda, x, \vec{y}, a, k, l)$ be the formula

$$(\forall i \leq x)\big(i \leq k \supset \lambda(i, \vec{y}) = \min\{a, t(i, \vec{y})\} \wedge \\ k \leq i < l \supset \lambda(i+1, \vec{y}) = \min\{h(i+1, \vec{y}, \lambda(i, \vec{y})), b(i+1, \vec{y})\}\big),$$

and let $D(x, \vec{y}, u)$ be the formula

$$(\forall a \leq t(x,\vec{y}))(\forall k \leq x)(\forall l \leq x)\big(0 \leq k \leq l \leq x \wedge l \dot{-} k \leq u \supset (\exists \lambda^t)C(\lambda^t, x, \vec{y}, a, k, l)\big).$$

We shall show that

$$U_2^1 \vdash D(x, \vec{y}, 0). \tag{2.1}$$

$$U_2^1 \vdash D(x, \vec{y}, u) \supset D(x, \vec{y}, 2u). \tag{2.2}$$

$$U_2^1 \vdash D(x, \vec{y}, u) \supset D(x, \vec{y}, 2u+1). \tag{2.3}$$

Since $D(x, \vec{y}, u)$ is a $\Sigma_1^{1,b}$-formula, (2.1)–(2.3) will allow us to use $\Sigma_1^{1,b}$-PIND to establish the existence condition for $f$.

For (2.1), observe that

$$U_2^1 \vdash D(x, \vec{y}, 0) \leftrightarrow \\ (\forall a \leq t(x, \vec{y}))(\forall k \leq x)(\exists \lambda^t)(\forall i \leq x)\big(i \leq k \supset \lambda^t(i, \vec{y}) = \min\{a, t(i, \vec{y})\}\big).$$

Therefore, (2.1) follows by an application of $\Delta_0^{1,b}$-comprehension.

To see (2.2), assume $D(x, \vec{y}, u)$ and let $a$, $k$, $l$ be such that

$$a \leq t(x, \vec{y}),\ k \leq x,\ l \leq x,\ 0 \leq k \leq l \leq x, l \dot{-} k \leq 2u.$$

If $k + u > x$ then $k + u > x \geq l$ whence $l \dot{-} k < u$. So, by hypothesis $D(x, \vec{y}, u)$ holds and (*a fortiori*) $D(x, \vec{y}, 2u)$ holds. If $k + u \leq x$ then by induction hypothesis,

$$U_2^1 \vdash \exists \lambda_1^t\, C(\lambda_1^t, x, \vec{y}, a, k, k+u).$$

Let $d = \lambda_1^t(k+u, \vec{y})$. Then we have

$$d = \lambda_1^t(k+u, \vec{y}) \leq b(k+u, \vec{y}) \leq t(k+u, \vec{y}) \leq t(x, \vec{y}),$$

because $t$ involves only the original function symbols of first order Bounded Arithmetic and $k + u \leq x$. By applying the induction hypothesis again we get

$$U_2^1 \vdash \exists \lambda_2^t\, C(\lambda_2^t, x, \vec{y}, d, k+u, l).$$

Now we use $\Delta_0^{1,b}$-comprehension to put $\lambda_1^t$ and $\lambda_2^t$ together:

$$U_2^1 \vdash (\exists\lambda^t)(\forall i \leq x)\big((i \leq k{+}u \supset \lambda^t(i) = \lambda_1^t(i)) \wedge (k{+}u{+}1 \leq i \leq l \wedge \lambda^t(i) = \lambda_2^t(i))\big).$$

That $C(\lambda^t, x, \vec{y}, a, k, l)$ holds is ensured by the following. For $i \leq x$,

$$\begin{aligned} i \leq k \;\supset\; & \lambda^t(i, \vec{y}) = \lambda_1^t(i, \vec{y}) = \min\{a, t(i, \vec{y})\}, \\ k \leq i < k+u \;\supset\; & \lambda^t(i+1, \vec{y}) = \lambda_1^t(i+1, \vec{y}) \\ & = \min\big\{h(i+1, \vec{y}, \lambda^t(i, \vec{y})), b(i+1, \vec{y})\big\}, \end{aligned}$$

$$\lambda^t(k{+}u{+}1, \vec{y}) = \lambda_2^t(k{+}u{+}1, \vec{y}) = \min\big\{h(k{+}u{+}1, \vec{y}, \lambda_2^t(k{+}u, \vec{y})), b(k{+}u{+}1, \vec{y})\big\},$$

but $\lambda_2^t(k{+}u, \vec{y}) = \min\big\{\lambda_1^t(k{+}u, \vec{y}), t(k{+}u, \vec{y})\big\} = \lambda_1^t(k{+}u, \vec{y}) = \lambda^t(k{+}u, \vec{y})$.

$$\begin{aligned} k+u+1 \leq i < l \;\supset\; & \lambda^t(i+1, \vec{y}) = \lambda_2^t(i+1, \vec{y}) \\ & = \min\big\{h(i+1, \vec{y}, \lambda_2^t(i, \vec{y})), b(i+1, \vec{y})\big\} \\ & = \min\big\{h(i+1, \vec{y}, \lambda^t(i, \vec{y})), b(i+1, \vec{y})\big\}. \end{aligned}$$

This shows that $U_2^1 \vdash \exists\lambda^t\, C(\lambda^t, x, \vec{y}, a, k, l)$, from which (2.2) follows.

To prove (2.3) we just need some minor modifications in the previous argument: assume again $D(x, \vec{y}, u)$ and let $a, k, l, u$ be such that $a \leq t(x, \vec{y})$, $k \leq x$, $l \leq x$, $0 \leq k \leq l \leq x$, $l \dot{-} k \leq 2u+1$, and $k+u \leq x$.

By induction hypothesis, $U_2^1$ proves that

$$\begin{aligned} & \exists\lambda_1^t\, C(\lambda_1^t, x, \vec{y}, a, k, k+u), \\ & \exists\lambda_2^t\, C(\lambda_2^t, x, \vec{y}, d, k+u, l \dot{-} 1), \\ & \exists\lambda_3^t\, C(\lambda_3^t, x, \vec{y}, e, l \dot{-} 1, l), \end{aligned}$$

where $d = \lambda_1^t(k+u, \vec{y})$ and $e = \lambda_2^t(l \dot{-} 1, \vec{y})$. Now we use $\Delta_0^{1,b}$-comprehension again to define $\lambda$:

$$\begin{aligned} U_2^1 \;\vdash\; (\exists\lambda^t)(\forall i \leq x)\big(i \leq k+u \;\supset\; \lambda^t(i) = \lambda_1^t(i) \wedge \\ (k+u+1 \leq i \leq l \dot{-} 1 \;\supset\; \lambda^t(i) = \lambda_2^t(i)) \;\wedge\; \lambda^t(l) = \lambda_3^t(l)\big). \end{aligned}$$

From this, $(\exists\lambda^t)\, C(\lambda^t, x, \vec{y}, a, k, l)$ will hold as before and (2.3) is proved. (Alternatively, we can get (2.3) by using (2.2) and showing that $U_2^1 \vdash D(x, \vec{y}, u) \supset D(x, \vec{y}, u+1)$. The proof of the latter is similar to that of (2.2) ).

By $\Sigma_1^{1,b}$-PIND it follows from (2.1)–(2.3) that $U_2^1 \vdash \forall u\, D(x, \vec{y}, u)$, whence $U_2^1 \vdash D(x, \vec{y}, x)$ because $x$ is free for $u$ in $D(x, \vec{y}, u)$. Since $x$, $\vec{y}$ are free variables we have $U_2^1 \vdash (\forall x)(\forall \vec{y})\, D(x, \vec{y}, x)$ by $\forall$-introduction, *i.e.*,

$$U_2^1 \vdash (\forall x)(\forall \vec{y})(\forall a \leq t(x,\vec{y}))(\forall k \leq x)(\forall l \leq x)\big(0 \leq k \leq l \leq x \;\wedge\; l \dot{-} k \leq x \\ \supset\; (\exists \lambda^t)\, C(\lambda^t, x, \vec{y}, a, k, l)\big).$$

If we set $a = \min\{g(\vec{y}), b(0, \vec{y})\}$, $l = x$, and $k = 0$ we will get

$$U_2^1 \vdash (\forall x)(\forall \vec{y})(\exists \lambda^t) C\big(\lambda^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x\big). \tag{2.4}$$

That is to say,

$$U_2^1 \vdash (\forall x)(\forall \vec{y})(\exists \lambda^t)(\forall i \leq x)\big(i = 0 \;\supset\; \lambda^t(i, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\} \\ \wedge\; 0 \leq i < x \;\supset\; \lambda^t(i+1, \vec{y}) = \min\{h(i+1, \vec{y}, \lambda^t(i, \vec{y})), b(i+1, \vec{y})\}\big). \tag{2.5}$$

This takes care of the existence part of the proof. We also need to prove that $\lambda^t$ is unique, that is, we need to prove that

$$U_2^1 \vdash C\big(\zeta^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x\big) \wedge \\ C\big(\theta^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x\big) \;\supset\; (\forall i \leq x)(\zeta^t(i, \vec{y}) = \theta^t(i, \vec{y})). \tag{2.6}$$

For that purpose, let $E(x, \vec{y}, \zeta^t, \theta^t)$ be the $\Delta_0^{1,b}$-formula

$$C\big(\zeta^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x\big) \;\wedge\; C\big(\theta^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x\big),$$

and let $F(u, x, \vec{y}, \zeta^t, \theta^t)$ be the $\Delta_0^{1,b}$-formula $(\forall i \leq u)(i \leq x \supset \theta^t(i, \vec{y}) = \zeta^t(i, \vec{y}))$. It is clear that

$$U_2^1 \vdash E(x, \vec{y}, \zeta^t, \theta^t) \;\supset\; F(0, x, \vec{y}, \zeta^t, \theta^t),$$
$$U_2^1 \vdash E(x, \vec{y}, \zeta^t, \theta^t) \;\wedge\; F(u, x, \vec{y}, \zeta^t, \theta^t) \;\supset\; F(u+1, x, \vec{y}, \zeta^t, \theta^t).$$

Since $U_2^1 \vdash \Delta_0^{1,b}$-IND, we have $U_2^1 \vdash E(x, \vec{y}, \zeta^t, \theta^t) \supset \forall u\, F(u, x, \vec{y}, \zeta^t, \theta^t)$, whence $U_2^1 \vdash E(x, \vec{y}, \zeta^t, \theta^t) \supset F(x, x, \vec{y}, \zeta^t, \theta^t)$, which is precisely statement (2.6).

If $A(x, \vec{y}, z)$ denotes the$\Sigma_1^{1,b}$-formula

$$\exists \lambda^t \,\big(C(\lambda^t, x, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, x) \;\wedge\; z = \lambda^t(x, \vec{y})\big),$$

then (2.4) and (2.6) show that $U_2^1 \vdash (\forall x)(\forall \vec{y})(\exists! z \leq t(x, \vec{y}))\, A(x, \vec{y}, z)$. Hence, according to definition 1.7, $A(x, \vec{y}, z)$ $\Sigma_1^{1,b}$-defines a function in $U_2^1$, and we can introduce a new function symbol $f$ and the defining axiom

$$f(x, \vec{y}) = z \leftrightarrow A(x, \vec{y}, z).$$

We can convince ourselves that

$$U_2^1 \vdash \forall \vec{y}\, f(0, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\}. \tag{2.7}$$

$$U_2^1 \vdash (\forall x)(\forall \vec{y}) f(x+1, \vec{y}) = \min\{h(x+1, \vec{y}, f(x, \vec{y})), b(x+1, \vec{y})\}. \tag{2.8}$$

Statement (2.7) follows immediately from

$$\begin{aligned} U_2^1 \vdash f(0, \vec{y}) = z &\leftrightarrow A(0, \vec{y}, z) \\ &\leftrightarrow \exists \lambda^t \left(C(\lambda^t, 1, \vec{y}, \min\{g(\vec{y}), b(0, \vec{y})\}, 0, 0) \wedge z = \lambda^t(0, \vec{y})\right) \\ &\leftrightarrow \exists \lambda^t \left(\lambda^t(0, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\} \wedge z = \lambda^t(0, \vec{y})\right). \end{aligned}$$

Statement (2.8) is easily obtained by using (2.7) and $\Delta_0^{1,b}$-IND on $x$ in the formula

$$f(x+1, \vec{y}) = \min\{h(x+1, \vec{y}, f(x, \vec{y})), b(x+1, \vec{y})\}.$$

Recall that $\Delta_0^{1,b}$-IND can be applied freely to $\Delta_0^{1,b}$-formulas containing $\Sigma_1^{1,b}$-defined function symbols.

From the second proof of the previous theorem we can also conclude the following.

**Proposition 2.3.** *Let $b$, $g$ and $h$ be $\Sigma_1^{1,b}$-definable functions in $U_2^1$, and let $t$ be a term of first order Bounded Arithmetic such that $U_2^1 \vdash b(x, \vec{y}) \leq t(x, \vec{y})$. If $E(\lambda, x, \vec{y})$ is the formula*

$$\begin{aligned} &(\forall i \leq x)\big(i = 0 \supset \lambda^t(i, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\} \wedge \\ &\quad \wedge\ 0 \leq i < x \supset \lambda^t(i+1, \vec{y}) = \min\{h(i+1, \vec{y}, \lambda^t(i, \vec{y})), b(i+1, \vec{y})\}\big), \end{aligned}$$

*then $U_2^1$ proves*

$$(\forall x)(\forall \vec{y})(\exists \lambda^t)\big(E(\lambda^t, x, \vec{y}) \wedge E(\zeta^t, x, \vec{y}) \wedge E(\theta^t, x, \vec{y}) \supset (\forall i \leq x)(\zeta^t(i, \vec{y}) = \theta^t(i, \vec{y}))\big)$$

*where $t$ is a term of first order Bounded Arithmetic such that*

$$U_2^1 \vdash (\forall x)(\forall \vec{y})(b(x, \vec{y}) \leq t(x, \vec{y})).$$

*Proof.* This is precisely the contents of statements (2.5) and (2.6) in the proof of Theorem 2.2.

**Definition 2.4.** *The second class of Grzegorczyk, $\mathcal{E}^2$, is the smallest class of number-theoretic functions containing $+$, $\cdot$, the constant functions and closed under substitution and bounded recursion.*

**Corollary 2.5.** *$U_2^1$ can $\Sigma_1^{1,b}$-define all $\mathcal{E}^2$ functions.*

*Proof.* This is immediate from Theorem 2.2.

**Corollary 2.6.** *$\mathcal{E}^2 \subseteq PSPACE$.*

*Proof.* This follows from Corollary 2.5 and Theorem 1.9.

**Corollary 2.7.** *The class of functions $\Sigma_1^{1,b}$-definable in $U_2^1$ is closed under bounded minimum, bounded maximum and bounded summation. That is to say, if $g(x, \vec{y})$ is $\Sigma_1^{1,b}$-definable in $U_2^1$, then the following functions are $\Sigma_1^{1,b}$-definable in $U_2^1$:*

$$(i) \quad f(x, \vec{y}) = \min_{i \leq x} g(i, \vec{y}).$$
$$(ii) \quad f(x, \vec{y}) = \max_{i \leq x} g(i, \vec{y}).$$
$$(iii) \quad f(x, \vec{y}) = \sum_{i \leq x} g(i, \vec{y}).$$

*Proof.* We can $\Sigma_1^{1,b}$-define these functions by using composition and bounded primitive recursion schemes as follows:

(*i*)
$$f(0, \vec{y}) = g(0, \vec{y}),$$
$$f(x+1, \vec{y}) = \min\{f(x, \vec{y}), g(x+1, \vec{y})\}.$$

(*ii*) First, observe that the function

$$h(x, \vec{y}) = \min_{i \leq x} [i \cdot \min_{j \leq x}(1 \dot{-} (g(j, \vec{y}) \dot{-} g(i, \vec{y})))]$$

is $\Sigma_1^{1,b}$-definable. Note that $h(x, \vec{y})$ is the smallest $i \leq x$ such that $(\forall j \leq x)(g(i, \vec{y}) \geq g(j, \vec{y}))$. So, $f(x, \vec{y})$ can be $\Sigma_1^{1,b}$-defined by $f(x, \vec{y}) = g(h(x, \vec{y}))$.

(*iii*)
$$f(0, \vec{y}) = g(0, \vec{y}),$$
$$f(x+1, \vec{y}) = \min\{ f(x, \vec{y}) + g(x+1, \vec{y}), (x+2) \max_{i \leq x+1} g(i, \vec{y}) \}.$$

Here the functions $h$ and $b$ of Theorem 2.2 are $h(x, \vec{y}, v) = g(x, \vec{y}) + v$ and $b(x, \vec{y}) = (x+1) \max_{i \leq x} g(i, \vec{y})$.

$U_2^1$ can also handle bounded recursion with cases as the next corollary shows.

**Corollary 2.8.** *If $b$, $g$, $h_1$ and $h_2$ are $\Sigma_1^{1,b}$-definable in $U_2^1$ and $A(x, \vec{y})$ is $\Delta_1^{1,b}$ with respect to $U_2^1$, then the function $f$ such that*

$$(\forall \vec{y})\ f(0, \vec{y}) = \min\{g(\vec{y}),\ b(0, \vec{y})\},$$
$$(\forall x)(\forall \vec{y}) f(x+1, \vec{y}) = \begin{cases} \min\{h_1(x+1, \vec{y}, f(x, \vec{y})), b(x+1, \vec{y})\}, & \text{if } A(x+1, \vec{y}) \\ \min\{h_2(x+1, \vec{y}, f(x, \vec{y})), b(x+1, \vec{y})\}, & \text{if } \neg A(x+1, \vec{y}) \end{cases}$$

*is $\Sigma_1^{1,b}$-definable in $U_2^1$.*

*Proof.* It is not difficult to see that $\chi_A(x, \vec{y})$ and $\chi_{\neg A}(x, \vec{y})$ are $\Sigma_1^{1,b}$-definable in $U_2^1$, so the result follows immediately from Theorem 2.2 by making

$$h(x, \vec{y}, z) = \chi_A(x, \vec{y}) h_1(x, \vec{y}, z) + \chi_{\neg A}(x, \vec{y}) h_2(x, \vec{y}, z).$$

The next proposition provides another type of recursion which can be carried out in $U_2^1$. On input $x$ this recursion clearly has depth $|x|$ which makes the proof very simple. As a matter of fact, proposition 2.10 below shows that in most instances this recursion can be performed via bounded recursion.

**Proposition 2.9.** *If $b$, $g$ and $h$ are $\Sigma_1^{1,b}$-definable in $U_2^1$, then the function $f$ such that*

$$(\forall \vec{y})\ f(0, \vec{y}) = \min\{g(\vec{y}),\ b(0, \vec{y})\},$$
$$(\forall x)(\forall \vec{y})\ (x \neq 0 \ \supset\ f(x, \vec{y}) = \min\{h(x, \vec{y}, f(\lfloor \tfrac{1}{2} x \rfloor, \vec{y})),\ b(x, \vec{y})\}$$

*is $\Sigma_1^{1,b}$-definable in $U_2^1$.*

*Proof.* We can give a direct proof in the theory $U_2^1$ similar to that of Theorem 2.2. However, this time we will be satisfied with presenting only the easiest argument. By Theorem 1.9 there are polynomials $q_g$, $q_b$, $q_h$; and $q_g$, $q_b$, $q_h$-SPACE bounded Turing machines computing $g$, $b$, $h$ (respectively). To compute $f(x, \vec{y})$ we apply the defining recursive scheme $|x|$ times. The result of each iteration is smaller than $b(x, \vec{y})$; therefore, for each $u < x$ the value $f(u, \vec{y})$ occupies less than $q_b(|u|, \vec{y})$ computation squares.Hence on input $x$, $\vec{y}$ the total computation space needed is *at most*

$$|x| + |\vec{y}| + q_g(|\vec{y}|) + |x|(q_h(|x|, |\vec{y}|, q_b(|x|, |\vec{y}|)) + q_b(|x|, |\vec{y}|)).$$

**Proposition 2.10.** *If $b$, $g$ and $h$ are $\Sigma_1^{1,b}$-definable in $U_2^1$, then the function $f$ such that*

$$\forall \vec{y}\ f(0, \vec{y}) = \min\{g(\vec{y}),\ b(0, \vec{y})\},$$
$$(\forall x)(\forall \vec{y})\ (x \neq 0 \ \supset\ f(x, \vec{y}) = \min\{h(\vec{y}, f(\lfloor \tfrac{1}{2} x \rfloor, \vec{y})), b(2^{|x|}, \vec{y})\}$$

*can be defined in $U_2^1$ by using bounded recursion.*

*Proof.* Let $b^*$ the function defined by

$$b^*(x, \vec{y}) = z \leftrightarrow A(x, \vec{y}, z)$$

where $A(x, \vec{y}, z)$ is the formula

$$(\exists u \leq x)(x = |u| + 1 \wedge z = b(2 \cdot 2^{|u|}, \vec{y})) \vee (\neg \exists u \leq x)(x = |u| + 1 \wedge z = 0).$$

It is clear that $A(x, \vec{y}, z)$ is a $\Sigma_1^{1,b}$-formula and that the existence and uniqueness conditions are satisfied; therefore $b^*$ is $\Sigma_1^{1,b}$-definable. By Theorem 2.2 the function $f^*$ such that

$$f^*(0, \vec{y}) = \min\{g(\vec{y}), b^*(0, \vec{y})\},$$
$$f^*(u+1, \vec{y}) = \min\{h(\vec{y}, f^*(u, \vec{y})), b(u+1, \vec{y})\}$$

is $\Sigma_1^{1,b}$-definable in $U_2^1$. Now $f$ can be defined by

$$f(0, \vec{y}) = \min\{g(\vec{y}), b(0, \vec{y})\},$$
$$x \neq 0 \supset f(x, \vec{y}) = f^*(|x|, \vec{y}).$$

Note that

$$\begin{aligned} u \neq 0 \supset f(2u, \vec{y}) = f(2u+1, \vec{y}) &= f^*(|u|+1, \vec{y}) \\ &= \min\{h(\vec{y}, f^*(|u|, \vec{y})), b^*(|u|+1, \vec{y})\} \\ &= \min\{h(\vec{y}, f(u, \vec{y})), b(2 \cdot 2^{|u|}, \vec{y})\}. \end{aligned}$$

Therefore, $x \neq 0 \supset f(x, \vec{y}) = \min\{h(\vec{y}, f(\lfloor \frac{1}{2}x \rfloor, \vec{y})), b(2^{|x|}, \vec{y})\}$.

In [4] we capitalize on the above results by showing that $U_2^1$ is strong enough to simulate relatively powerful combinatorial arguments.

## References


[1] D. ANGLUIN, *Lecture notes on the complexity of some problems in number theory*, Technical Report 234, Yale University, August 1982.

[2] S. BUSS, *Bounded Arithmetic*, Bibliopolis, Napoli, Italia, 1986.

[3] J. L. BALCAZAR, J. DIAZ & J. GABARRO, *Structural Complexity I*, Springer-Verlag, New York, 1988.

[4] R. DE CASTRO, *General Combinatorial Principles in Second Order Bounded Arithmetic*, Lecture Notes in Pure and Applied Mathematics, Vol. 203, Marcel Dekker, 1998, 373-395.

[5] A. J. MACINTYRE, *The strength of weak systems*, Proceedings of the 11th International Wittgenstein Symposium, Kirchberg-Wechsel, Austria, 1986, 43–59.

[6] G. TAKEUTI, *Proof Theory, 2nd ed.*, North-Holland, Amsterdam, 1980.

[7] A. YASUHARA, *Recursive Function Theory and Logic*, Academic Press, 1971.