Portugal. Math. (N.S.)
Vol. xx, Fasc. , 200x, xxx–xxx

Portugaliae Mathematica

Bounded theories for polyspace computability

Ricardo Bianconi, Gilda Ferreira, Emmanuel Silva*

Abstract. We present theories of bounded arithmetic and weak analysis whose provably total functions (with appropriate graphs) are the polyspace computable functions. More precisely, inspired in Ferreira's systems PTCA, Σ^b_1-NIA and BTFA in the polytime framework, we propose analogue theories concerning polyspace computability. Since the techniques we employ in the characterization of PSPACE via formal systems (e.g. Herbrand's theorem, cut-elimination theorem and the expansion of models) are similar to the ones involved in the polytime setting, we focus on what is specific of polyspace and explains the lift from PTIME to PSPACE.

Mathematics Subject Classification (2010). Primary 68Q15, 03F30, 03F35, 03D15; Secondary 03F03, 03F05.

Keywords. Bounded arithmetic. Weak analysis. Polyspace computability. Conservation results. Cut-elimination.

1. Introduction

Close connections between bounded theories of arithmetic and computational complexity classes have been established in the mid 1980's by Samuel Buss. In [2], Buss introduces the weak formal systems S^1_2, U^1_2 and V^1_2, whose provably total functions (with appropriate graphs) are respectively the classes PTIME, PSPACE and EXPTIME. The idea is that weak (subexponential) theories can be used to analyze complexity-theoretic questions. For more on *bounded arithmetic* and related work see [21, 20, 22, 4, 18].

A few years later [8, 9], Fernando Ferreira presents alternative characterizations of polytime computability via formal systems. Among the systems proposed we highlight the theory Σ^b_1-NIA, which corresponds to Buss's system S^1_2 in a binary notation framework (see [17]), and the system PTCA which allows induction for polynomial time decidable predicates.

*The second author acknowledges support of FCT-Fundação para a Ciência e a Tecnologia (project PTDC/MAT/104716/2008 and grant SFRH/BPD/34527/2006), Centro de Matemática e Aplicações Fundamentais da Universidade de Lisboa (PEst-OE/MAT/UI0209/2011) and is also grateful to Núcleo de Investigação em Matemática da Universidade Lusófona.

Following this line of research (weak theories in binary notation characterizing polytime computability) and responding to a challenge of Wilfried Sieg (find a subsystem for analysis whose provably recursive functions consists only of the computationally feasible ones), Ferreira introduces in [10] the theory BTFA. BTFA is proposed as a base theory in the *reverse mathematics*'s style, playing, in the polytime framework, the role that RCA_0 plays in the original reverse mathematic's setting (see [26]). While PTCA and Σ_1^b-NIA are first-order theories, since BTFA is designed to the formalization of analysis, it is a second-order system, being able to deal not only with binary words but with sets of words. For the formalization of analysis in BTFA see [6].

In the present paper we focus in polyspace computability. The goal is to present theories of bounded arithmetic and weak analysis whose provably total functions (with appropriate graphs) are the polyspace computable functions. Since the study in the polyspace framework is strongly inspired and guided by Ferreira's work in the polytime setting, we give special emphasis to what is specific of PSPACE and refer to Ferreira's work in the common parts.

The paper is organized as follows.

In Section 2, we recall a recursion-theoretic characterization of PSPACE and introduce the theory PSCA proving, via Herbrand's theorem, that its provably total functions are the polyspace computable functions. The idea is that PSCA follows the informal correspondence:

$$\frac{\mathsf{PTCA}}{\mathsf{PTIME}} \sim \frac{\mathsf{PSCA}}{\mathsf{PSPACE}}.$$

In the next section, we introduce a second system designed to capture polyspace computability, $\Sigma_1^{1,b}$-NIA. The idea is to avoid the need for function symbols for each polyspace computable function (as in PSCA), reproducing in the new setting the simplification of language achieved in the polytime framework with the move from PTCA to Σ_1^b-NIA. Graphically, we have the correspondence:

$$\frac{\Sigma_1^b\text{-}\mathsf{NIA}}{\mathsf{PTCA}} \sim \frac{\Sigma_1^{1,b}\text{-}\mathsf{NIA}}{\mathsf{PSCA}}.$$

While Σ_1^b-NIA is a first-order theory, the theory $\Sigma_1^{1,b}$-NIA is a second-order bounded system. As discussed in the section, the latter theory has to be able to deal with (bounded) sets of words in order to capture functions described by bounded recursion. Note that the class PSPACE is closed under the bounded recursion scheme. The strategy to prove that $\Sigma_1^{1,b}$-NIA corresponds to polyspace computability is similar to the one used by Buss to prove an homologous result (in unary notation) relatively to U_2^1: an application of partial cut-elimination.

The enrichment of $\Sigma_1^{1,b}$-NIA with a bounded collection scheme keeping the connection with polyspace computability is done in Section 4 also via proof-theoretic means. The importance of the new scheme becomes visible in the subsequent section when joining recursive comprehension to the system: a fundamental scheme to the development of analysis.

In Section 5, we finally present a theory rich enough to develop analysis keeping the connection with polyspace computability. We named the theory BTPSA and were inspired in the following relation:

$$\frac{\mathsf{BTFA}}{\Sigma^b_1\text{-}\mathsf{NIA}} \sim \frac{\mathsf{BTPSA}}{\Sigma^{1,b}_1\text{-}\mathsf{NIA}}.$$

The principal features of BTPSA (and of other weak theories for analysis) are a second-order language (with unbounded second-order variables) and a recursive comprehension scheme, central in expressing, for instance, the existence of particular real numbers and in the formalization of basic analytic concepts such as continuous real functions or sequences of real numbers.

In the recent paper [7], a general blueprint for the construction of theories for analysis connected with computational complexity classes is presented and is illustrated in the polytime framework. This work can be seen as the implementation of the general blueprint in the polyspace setting.

The results of this paper are part of the master thesis [15]. The material in Section 2 was, in addition to [15], also presented (in a slight different formulation) in the PhD dissertation [24]. Since both dissertations are written in Portuguese, we found it useful to make the work visible to the logic community through this article.

2. A first-order arithmetic theory for polyspace computability

In this section, we start by recalling an alternative characterization of PSPACE, the well-known computational complexity class usually defined by limiting the amount of space available (polynomial space) in a deterministic Turing Machine. The inductive characterization of PSPACE we present, is essentially the one introduced in [23] and will be useful in what follows. Since it is written in binary notation, we start introducing some operations. Let $2^{<\omega}$ (also known as $\{0,1\}^*$) be the set of all finite sequences of 0's and 1's. The empty sequence is denoted by ϵ. For x and y elements in $2^{<\omega}$, $x\hat{\ }y$ represents the *concatenation* of x by y (we usually omit the symbol $\hat{\ }$ and just write xy); $x \subseteq y$ means that x is an *initial subword* of y (string prefix); $|x|$ denotes the *length* of x, i.e. the number of 0's and 1's in the word x; $x_{|_y}$ is the *truncation* of x by y defined by $x_{|_y} := \begin{cases} x, & \text{if } |x| \leq |y| \\ z, & \text{if } z \subseteq x \wedge |z| = |y| \end{cases}$; $x \times y$ is the *product* of x by y defined as being the word x concatenated with itself length of y times; $x \preceq y$ (respectively $x \equiv y$) abbreviates $1 \times x \subseteq 1 \times y$ (respectively $1 \times x = 1 \times y$) meaning that the length of x is less than or equal (respectively equal) to the length of y; and $\leq_l$ is the linear order defined by $x \leq_l y :\Leftrightarrow (x \preceq y \wedge \neg(x \equiv y)) \vee (x \equiv y \wedge \exists z \subseteq x\,(z0 \subseteq x \wedge z1 \subseteq y)) \vee (x = y)$, i.e. it is defined first according to length and then, within the same length, lexicographically.

Definition 2.1. PSPACE is the smallest class of functions that includes the initial functions:

(1) $C_0(x) = x0$

(2) $C_1(x) = x1$

(3) $P^n_i(x_1, ..., x_n) = x_i$, for $1 \le i \le n$

(4) $Q(x, y) = \begin{cases} 1, & \text{if } x \subseteq y \\ 0, & \text{otherwise} \end{cases}$

and is closed under the following schemes:

- composition

  $f(\bar{x}) = g(h_1(\bar{x}), ..., h_k(\bar{x}))$

- bounded recursion on notation

  $f(\bar{x}, \epsilon) = g(\bar{x})$

  $f(\bar{x}, y0) = h_0(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$

  $f(\bar{x}, y1) = h_1(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$,

  where t is a bounding function[1], i.e. t belongs to the smallest class of functions that includes $\epsilon, 0, 1, ^\frown, \times, P^n_j$ and is closed under composition

- bounded recursion

  $f(\bar{x}, \epsilon) = g(\bar{x})$

  $f(\bar{x}, S(y)) = h(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$,

  where t is a bounding function and S is the successor function defined by $S(\epsilon) = 0, S(x0) = x1, S(x1) = S(x)0$.

Note that the last scheme is essential to capture polyspace computability. Removing the scheme of bounded recursion in the definition above we obtain exactly a characterization of PTIME. See [8].

Let $\mathcal{L}$ be the first-order language which has three constant symbols ϵ, 0 and 1, two binary function symbols $^\frown$ and $\times$ (intended to be interpreted respectively as *concatenation* and *product* in the standard model) and two binary relation symbols $=$ and $\subseteq$ (for *equality* and *initial subwordness* respectively). The domain of the intended standard model of the language is $2^{<\omega}$. Let $\mathcal{L}_{\mathcal{PS}}$ be an extension of the former language by adding a function symbol for each description of a polyspace computable function according to Definition 2.1.

Definition 2.2. The class of *polyspace decidable matrices* is the smallest class of formulas of $\mathcal{L}_{\mathcal{PS}}$ containing the atomic formulas and closed under the Boolean operations and quantifications of the form $\forall x\, (x \preceq t \to \ldots)$ or $\exists x\, (x \preceq t \wedge \ldots)$, where t is a term of $\mathcal{L}_{\mathcal{PS}}$ where x does not occur.

Definition 2.3. PSCA (acronym for *Polynomial Space Computable Arithmetic*) is the first-order theory, in the language $\mathcal{L}_{\mathcal{PS}}$, which has the following axioms:

- *Basic axioms*

[1] The bounding functions ensure that the recursion scheme does not produce functions with exponential growth.

$x\epsilon = x$, $x(y0) = (xy)0$ *and* $x(y1) = (xy)1$;

$x \times \epsilon = \epsilon$, $x \times y0 = (x \times y)x$ *and* $x \times y1 = (x \times y)x$;

$x \subseteq \epsilon \leftrightarrow x = \epsilon$, $x \subseteq y0 \leftrightarrow x \subseteq y \vee x = y0$ *and* $x \subseteq y1 \leftrightarrow x \subseteq y \vee x = y1$;

$x0 = y0 \rightarrow x = y$ *and* $x1 = y1 \rightarrow x = y$;

$x0 \neq y1$, $x0 \neq \epsilon$ *and* $x1 \neq \epsilon$;

- *Defining axioms*

  a. Initial functions

  (1) $C_0(x) = x0$
  (2) $C_1(x) = x1$
  (3) $P_i^n(x_1, ..., x_n) = x_i$, for $1 \leq i \leq n$
  (4) $Q(x, y) = 1 \leftrightarrow x \subseteq y$; $Q(x, y) = 0 \vee Q(x, y) = 1$

  b. Derived functions

  (1) $f(\bar{x}) = g(h_1(\bar{x}), ..., h_k(\bar{x}))$,
  if f is the description of the composition from $g, h_1, \ldots, h_k$
  (2) $f(\bar{x}, \epsilon) = g(\bar{x})$
  $f(\bar{x}, y0) = h_0(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$
  $f(\bar{x}, y1) = h_1(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$,
  where t is a term of the language $\mathcal{L}$ and f is the description of the bounded recursion on notation defined from g, h_0, h_1 and t
  (3) $f(\bar{x}, \epsilon) = g(\bar{x})$
  $f(\bar{x}, S(y)) = h(\bar{x}, y, f(\bar{x}, y))_{|_{t(\bar{x},y)}}$,
  where t is a term of $\mathcal{L}$, S is the successor function and f is the description of the bounded recursion defined from g, h and t

- *Scheme of induction on notation*

  $A(\epsilon) \wedge \forall x\, (A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow \forall x\, A(x)$,

  where A is a polyspace decidable matrix, possibly with other free variables besides x.

The theory PSCA described above was inspired in PTCA (*Polynomial Time Computable Arithmetic*), a theory introduced by Ferreira in [8] designed to correspond to polytime computability. The novelty in PSCA is that the scope of the induction on notation scheme extends from polytime decidable matrices to *polyspace* decidable matrices and in the derived functions we add an extra scheme for *bounded recursion*.

Note that the scheme of *slow induction*

$$A(\epsilon) \wedge \forall x\, (A(x) \rightarrow A(S(x))) \rightarrow \forall x\, A(x)$$

(which in unary notation corresponds to the usual +1-induction) is valid in PSCA for A a polyspace decidable matrix possibly with other free variables besides x. The proof is entirely similar to the one presented in [8], page 53.

Having in view to show that PSCA corresponds to the computational complexity class PSPACE, i.e. to polyspace computability, we start by arguing that PSCA is a universal theory. Since it is not visible from the above formulation because of the induction scheme, our strategy is to use a well-known result of Łoś and Tarski that ensures that if a theory is preserved by substructures then it is a universal theory.

In order to prove that PSCA is preserved by substructures we need some auxiliary results.

Lemma 2.4. *Given $A_1(\bar{x}), \ldots, A_n(\bar{x})$ polyspace decidable matrices and $f_1(\bar{x}), \ldots, f_{n+1}(\bar{x})$ function symbols, there is a function symbol $f(\bar{x})$ such that*
$\mathsf{PSCA} \vdash (A_1(\bar{x}) \wedge f(\bar{x}) = f_1(\bar{x})) \vee (\neg A_1(\bar{x}) \wedge A_2(\bar{x}) \wedge f(\bar{x}) = f_2(\bar{x})) \vee \ldots \vee (\neg A_1(\bar{x}) \wedge \ldots \wedge \neg A_n(\bar{x}) \wedge f(\bar{x}) = f_{n+1}(\bar{x}))$.

The result above shows that PSCA allows the definition of functions by cases, being the cases expressed by polyspace decidable matrices.

Proof. With the extra assumption that for the polyspace decidable matrices A_i there are function symbols K_{A_i} such that $\mathsf{PSCA} \vdash (A_i(\bar{x}) \to K_{A_i}(\bar{x}) = 1) \wedge (\neg A_i(\bar{x}) \to K_{A_i}(\bar{x}) = 0)$, Lemma 2.4 can easily be proved by induction on n. For $n = 1$ just take $f(\bar{x}) = h(f_2(\bar{x}), f_1(\bar{x})K_{A_1}(\bar{x}))$ with $h(y, \epsilon) = y$, $h(y, x0) = y$ and $h(y, x1) = x$.

Thus, it just remains to argue that polyspace decidable matrices can be expressed in PSCA by means of quantifier-free formulas, more precisely for each polyspace decidable matrix A there is a function symbol K_A in $\mathcal{L}_{\mathcal{PS}}$ such that $\mathsf{PSCA} \vdash (A(\bar{x}) \to K_A(\bar{x}) = 1) \wedge (\neg A(\bar{x}) \to K_A(\bar{x}) = 0)$.

The proof can be done by induction on the complexity of A. For A an atomic formula we take $K_{\subseteq}$ as being Q and $K_{=}(x, y)$ as being $Q(11, Q(x, y)Q(y, x))$. We define $K_{\neg B} :\equiv K_{=}(0, K_B)$ and $K_{B\wedge C} :\equiv K_{=}(11, K_B K_C)$. Let $A(\bar{z}, x)$ be the formula $\forall y\,(y \preceq x \to B(\bar{z}, y))$. Take $f(\bar{z}, \epsilon) = K_B(\bar{z}, \epsilon)$ and

$$f(\bar{z}, S(x)) = \begin{cases} f(\bar{z}, x) & \text{if } K_B(\bar{z}, S(x)) = 1 \\ 0 & \text{if } K_B(\bar{z}, S(x)) = 0. \end{cases}$$

By slow induction on x, it is easy to prove that $\forall y\,(y \leq_l x \to B(\bar{z}, y)) \leftrightarrow f(\bar{z}, x) = 1$. So, since $y \preceq x \leftrightarrow y \leq_l 1 \times x$, we have $\forall y\,(y \preceq x \to B(\bar{z}, y)) \leftrightarrow f(\bar{z}, 1 \times x) = 1$. The result follows taking $K_A(\bar{z}, x)$ as being $K_{=}(1, f(\bar{z}, 1 \times x))$.

□

Lemma 2.5. *For each polyspace decidable matrix $A(\bar{z}, x)$ there is a function symbol g in $\mathcal{L}_{\mathcal{PS}}$ such that* $\mathsf{PSCA} \vdash (\exists y \preceq x\, A(\bar{z}, y)) \to g(\bar{z}, x) \preceq x \wedge A(\bar{z}, g(\bar{z}, x))$.

Proof. Take $f(\bar{z}, \epsilon) = \begin{cases} \epsilon & \text{if } A(\bar{z}, \epsilon) \\ 1 & \text{otherwise} \end{cases}$

$$f(\bar{z}, S(y)) = \begin{cases} f(\bar{z}, y) & \text{if } f(\bar{z}, y) \preceq y \\ S(y) & \text{if } f(\bar{z}, y) \npreceq y \text{ and } A(\bar{z}, S(y)) \\ S(y)1 & \text{if } f(\bar{z}, y) \npreceq y \text{ and } \neg A(\bar{z}, S(y)). \end{cases}$$

As a bounding function we can take $1 \times y11$. Let $g(\bar{z}, x)$ be (by definition) $f(\bar{z}, 1 \times x)$. Let us prove that g satisfies the statement in the lemma. It can be proved, by slow induction on y that $f(\bar{z}, y) \preceq y \to A(\bar{z}, f(\bar{z}, y))$, thus we have that (†) $g(\bar{z}, y) \preceq y \to A(\bar{z}, g(\bar{z}, y))$. Also, by slow induction, this time on x, we can prove that $(\exists y \leq_l x\, A(\bar{z}, y)) \to f(\bar{z}, x) \leq x$. Easily from the above statement we can prove $(\exists y \leq x\, A(\bar{z}, y)) \to g(\bar{z}, x) \leq x$. The result follows from the last assertion and (†). □

Let $\mathcal{M}$ be a model of PSCA and $\mathcal{N}$ a substructure of $\mathcal{M}$. From Lemma 2.5 we can easily argue that the polyspace decidable matrices are absolute between $\mathcal{N}$ and $\mathcal{M}$. The argument used in [9] page 143, in the polytime setting, adapts trivially to the present context. To conclude that PSCA is preserved by substructures, we only need to check that induction on notation also holds in $\mathcal{N}$. Since the induction scheme can be reformulated in the following alternative way:

$$A(\epsilon) \wedge \forall x \subseteq a\, (A(x) \to A(x0) \wedge A(x1)) \to A(a),$$

by absoluteness of the polyspace decidable matrices, the induction on notation axioms hold in $\mathcal{N}$.

Therefore, since PSCA is a universal theory, applying the Herbrand Theorem and Lemma 2.4, immediately we conclude that:

Theorem 2.6. *If* PSCA $\vdash \forall \bar{x} \exists y A(\bar{x}, y)$ *with A a polyspace decidable matrix and $\bar{x}$ and y its only free variables then, there is a function symbol f in $\mathcal{L}_{\mathcal{PS}}$ such that* PSCA $\vdash \forall \bar{x} A(\bar{x}, f(\bar{x}))$.

It is in this precise sense - the provably total functions of PSCA with polyspace graphs are exactly the functions of PSPACE - that we say that PSCA corresponds to polyspace computability.

3. A second-order bounded arithmetic theory for polyspace computability

It is known that it is possible to introduce all primitive recursive functions in Σ^0_1-IND (see the incompleteness paper of Gödel) and it is possible to introduce all the polytime computable functions in Σ^b_1-NIA (see [9]). As a consequence, PRA can be consider a subtheory of Σ^0_1-IND and PTCA a subtheory of Σ^b_1-NIA.

The goal of this section is to present a theory which plays, in the polyspace setting, the role played by Σ^0_1-IND and Σ^b_1-NIA in the primitive recursive and polytime settings respectively. More precisely, we are looking for a theory still characterizing polyspace computability but in a more economic language not having all descriptions of PSPACE functions as primitive.

While Σ^b_1-NIA corresponds to Buss's system S^1_2, the theory $\Sigma^{1,b}_1$-NIA we are going to introduce is inspired in Buss's system U^1_2. Contrarily to what happens with polytime computability (where a first-order language is enough), to introduce functions described by bounded recursion we need $\Sigma^{1,b}_1$-NIA to be a second-order (bounded) theory. Note that, in a system that does not prove the totality of exponentiation, not every bounded set is given by a binary word. The introduction of second-order bounded variables is by no means a novelty, Buss in [2] uses second-order bounded variables in the unary notation context.[2]

We start introducing the second-order language and some notation.

Let $\mathcal{L}_2^b$ be the second-order language obtained from $\mathcal{L}$ by adding second-order bounded variables and the relation symbol $\in$ which infixes between a term of $\mathcal{L}$ and a second-order bounded variable. The *standard* structure for this language has domain $(2^{<\omega}, \mathcal{P}_f(2^{<\omega}))$, i.e., the first-order variables are interpreted as finite sequences of zeros and ones, and the second-order (bounded) variables are subsets X^t of $2^{<\omega}$, with t any term of $\mathcal{L}$, satisfying $x \in X^t \rightarrow x \leq t$.

The *terms* in $\mathcal{L}_2^b$ coincide with the terms in $\mathcal{L}$ and the *class of formulas* in $\mathcal{L}_2^b$ can be defined as the smallest class of expressions containing the atomic formulas $t_1 \subseteq t_2, t_1 = t_2, t_1 \in X^t$, and closed under the Boolean operations, the first-order quantifications $\forall x, \exists x$, the first-order bounded quantifications $\forall x \leq t, \exists x \leq t$ and the second-order bounded quantifications $\forall X^t, \exists X^t$. Note that in $\mathcal{L}_2^b$, $\forall x \leq t\, A$ and $\exists x \leq t\, A$ are treated as new formulas and not as mere abbreviations for $\forall x\, (x \leq t \rightarrow A)$ and $\exists x\, (x \leq t \wedge A)$ respectively. It is a technical detail that contributes for an efficient formulation of sequent calculus.

A $\Sigma_1^{1,b}$-*formula* (respectively $\Pi_1^{1,b}$-*formula*) is a formula in the language $\mathcal{L}_2^b$ of the form: $\exists X_1^{t_1} ... \exists X_k^{t_k}\, A$ (respectively $\forall X_1^{t_1} ... \forall X_k^{t_k}\, A$), where A is a $\Sigma_0^{1,b}$-formula (i.e. with no quantifications of second-order and where all the first-order quantifications are bounded. It may have first and second-order parameters). In the standard model, if the second-order parameters are in the Polynomial Hierarchy (a.k.a. Meyer-Stockmeyer Hierarchy) then the $\Sigma_0^{1,b}$-formulas define predicates in this hierarchy. An *extended* $\Sigma_1^{1,b}$-*formula* (respectively *extended* $\Pi_1^{1,b}$-*formula*) is a formula that can be built in a finite number of steps, starting with $\Sigma_0^{1,b}$-formulas and allowing conjunctions, disjunctions, first-order bounded quantifications and second-order bounded existential (respectively universal) quantifications.

Definition 3.1. $\Sigma_1^{1,b}$-NIA is the second-order theory in the language $\mathcal{L}_2^b$, which has the following axioms:

- *Basic axioms* [3],
- $\forall x \forall X^t (x \in X^t \rightarrow x \leq t)$, with t a term where x does not occur,
- *Bounded comprehension*: $\exists X^t \forall x \leq t(x \in X^t \leftrightarrow A(x))$, where t is a term in which x does not occur, and A is a $\Sigma_0^{1,b}$-formula that may have other free variables other than x and where the variable X^t does not occur,
- *Induction on notation for* $\Sigma_1^{1,b}$-*formulas*:

$$A(\epsilon) \wedge \forall x\, (A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow \forall x\, A(x),$$

  with A a $\Sigma_1^{1,b}$-formula possibly with free variables other than x,

[2] We thank the anonymous referee for calling our attention to two related works: (i) Skelley [25] gives a characterization of PSPACE via a third-order (bounded) theory; (ii) Kołodziejczyk, Nguyen and Thapen [19] use a strategy similar to ours to express PSPACE computability in their framework.

[3] The 14 basic axioms of Definition 2.3.

- *Replacement for $\Sigma_0^{1,b}$-formulas*: $\forall x \leq t \exists X^q A(x, X^q) \to \exists Y^r \forall x \leq t \bar{A}(x, Y^r)$, with A a $\Sigma_0^{1,b}$-formula, t a term where x does not occur, and $\bar{A}$ results from A by replacing all the occurrences of '$s \in X^q$' by '$\langle x, s\rangle \in Y^r$' (where $\langle\,,\,\rangle$ is a pairing function and r is a certain term depending on t and q). We are omitting the exact term r in order to facilitate reading (the term depends on the particular definition of the pairing function – see [15] for a concrete implementation of these matters). This is a technical axiom that permits a kind of "permutation" between bounded first-order universal quantifications and bounded second-order existential quantifications.

Applying the replacement scheme, it is straightforward to see that an extended $\Sigma_1^{1,b}$-formula can be expressed via a $\Sigma_1^{1,b}$-formula. Thus, contrarily to the formulation of Buss's theory U_2^1, where the presence of *extended* $\Sigma_1^{1,b}$-formulas in the induction scheme is essential, with the replacement scheme available we can disregard extended formulas in the formulation of $\Sigma_1^{1,\mathsf{b}}$-NIA, simplifying subsequent arguments by cut-elimination.

Later, in Section 5, we will need a stronger form of comprehension still available in $\Sigma_1^{1,\mathsf{b}}$-NIA. More precisely, we will use the fact that

$$\forall x(A(x) \leftrightarrow B(x)) \to \forall w \exists X^w \forall x \leq w(x \in X^w \leftrightarrow A(x)),$$

with A an extended $\Sigma_1^{1,b}$-formula and B an extended $\Pi_1^{1,b}$-formula, is derivable in $\Sigma_1^{1,\mathsf{b}}$-NIA. We call the previous scheme $\Delta_1^{1,b}$- *bounded comprehension*. The details of this quite technical result are available in [15], pages 32–34.

Next we argue that PSCA is a subtheory of $\Sigma_1^{1,\mathsf{b}}$-NIA.

It is possible to present for each $f(\overline{x}) \in \mathcal{L}_{\mathcal{PS}} \backslash \mathcal{L}$ a $\Sigma_1^{1,b}$-formula $F_f(\overline{x}, y)$ in the language $\mathcal{L}_2^b$ and a term b_f in $\mathcal{L}$ such that $\Sigma_1^{1,\mathsf{b}}$-$\mathsf{NIA} \vdash \forall \overline{x}\, \exists^1 y \leq b_f(\overline{x})\, F_f(\overline{x}, y)$ and which has the defining properties of f (as given by Definition 2.3)[4]. The proof is by induction on the complexity of the description of f. If f is an initial function or is defined by composition or bounded recursion on notation see the proof presented in [9], pages 150–151, in the context of PTIME. What is new, concerning PSPACE, is the bounded recursion scheme. It remains to prove that if f is the description of bounded recursion defined from g, h and t, there is a $\Sigma_1^{1,b}$-formula $F_f(\overline{x}, y, z)$ and a term $b_f(\bar{x}, y)$ in the conditions above satisfying, in particular, $\Sigma_1^{1,\mathsf{b}}$-$\mathsf{NIA} \vdash F_g(\overline{x}, z) \to F_f(\overline{x}, \epsilon, z)$ and $\Sigma_1^{1,\mathsf{b}}$-$\mathsf{NIA} \vdash F_f(\overline{x}, y, r) \wedge F_h(\overline{x}, y, r, u) \wedge z = u_{|_{t(\overline{x},y)}} \to F_f(\overline{x}, S(y), z)$. The details can be found in [15], pages 35–40. Here we just give some intuition. In the case of bounded recursion on notation, to express the value of the function on y, we need the values of the function on the initial subwords of y (thinking in terms of binary trees we are going through a path till reaching the initial node). With appropriate coding, it is possible to collect all this information in a word bounded by a term (depending on y and on the bounding term t of the function). However, in the case of bounded recursion, the above strategy is no longer possible since to determine the value of the function on y, we need the values of the function in all words $x \leq_l y$ (in terms of binary trees we are going down the words of the same length starting in y till reaching the initial node). An attempt to code the information as before would result on exponential size. This is the reason why $\Sigma_1^{1,\mathsf{b}}$-NIA is formulated in a language expressive enough to denote

[4] By $\exists^1$ we mean "there is one and only one", i.e. $\Sigma_1^{1,\mathsf{b}}$-$\mathsf{NIA} \vdash F_f(\overline{x}, y) \wedge F_f(\overline{x}, z) \to y = z$.

bounded sets instead of just words. See [15], page 37, for a possible way of collecting all the information on a polynomially bounded set (the set allows the generation of a sequence of blocks, coding exactly the ordered values of the function, by generating 1 or 0 depending on the fact that a word is or is not in the set) and for the precise construction of the $\Sigma_1^{1,b}$-formula F_f and term b_f, with f a function obtained by bounded recursion from g, h and t. Informally, $F_f(\bar{x}, y, z)$ will say that there exists a bounded set which generate the blocks as above and such that if r is the yth block, then z is coded in r.

The proof that F_f and b_f satisfy the expected conditions uses the fact that in $\Sigma_1^{1,\mathsf{b}}$-NIA is valid slow induction for $\Sigma_0^{1,b}$-formulas. See the details in [15], pages 33–40.

We can subsume the above discussion - PSCA as a subtheory of $\Sigma_1^{1,\mathsf{b}}$-NIA - by the following result:

Proposition 3.2. *Every model $\mathcal{M}$ of $\Sigma_1^{1,\mathsf{b}}$-NIA can be extended to a model $\mathcal{M}'$ of* PSCA *keeping the first-order domain and the interpretations of symbols of $\mathcal{L}$ and defining the interpretation of each function symbol $f \in \mathcal{L}_{\mathcal{PS}} \backslash \mathcal{L}$ via F_f, i.e., interpreting f as the function $\{(\bar{a}, b) : \mathcal{M} \models F_f(\bar{a}, b)\}$.*

Proof. From the study above, by the definition of $\mathcal{M}'$ and of the formulas F_f for every $f \in \mathcal{L}_{\mathcal{PS}} \backslash \mathcal{L}$, the basic and defining axioms hold in $\mathcal{M}'$. For the remaining case, the induction scheme, it is enough to show that for every polyspace decidable matrix $A'(\bar{x})$ there is an extended $\Sigma_1^{1,b}$-formula $A(\bar{x})$ such that $\mathcal{M}' \models A'(\bar{a})$ iff $\mathcal{M} \models A(\bar{a})$, for every tuple of parameters $\bar{a}$ in $\mathcal{M}'$ (see the consideration concerning extended formulas in the beginning of the section). By induction on the complexity of the term, the existence of the previous $\Sigma_1^{1,b}$-formulas of $\mathcal{L}_2^b$ and terms of $\mathcal{L}$ can be extended from function symbols of $\mathcal{L}_{\mathcal{PS}} \backslash \mathcal{L}$ to every term t of $\mathcal{L}_{\mathcal{PS}}$ having that $\mathcal{M}' \models t(\bar{a}) = b$ iff $\mathcal{M} \models F_t(\bar{a}, b)$. Think in A' (modulo equivalence of formulas) as containing a sequence (possible empty) of quantifications of the form $\forall x \leq t(\bar{y})$ and $\exists x \leq t(\bar{y})$ followed by a quantifier-free formula in the conjunctive normal form. The construction of A from A' is done in the expected way. For example, if in A' appears the atomic formula $t(\bar{x}) \subseteq q(\bar{x})$ with t and q terms of $\mathcal{L}_{\mathcal{PS}}$, in A we will have $\exists z \leq b_t(\bar{x}) \exists w \leq b_q(\bar{x})(F_t(\bar{x}, z) \wedge F_q(\bar{x}, w) \wedge z \subseteq w)$. Negations of atomic formulas, like $t(\bar{x}) \neq q(\bar{x})$ in A' will be expressed in A by $\exists z \leq b_t(\bar{x}) \exists w \leq b_q(\bar{x})(F_t(\bar{x}, z) \wedge F_q(\bar{x}, w) \wedge z \neq w)$. And quantifications of the form $\forall x \leq t(\bar{y}) \ldots$ or $\exists x \leq t(\bar{y})$ are replaced by $\exists z \leq b_t(\bar{y})(F_t(\bar{y}, z) \wedge \forall x \leq z \ldots)$ or $\exists z \leq b_t(\bar{y})(F_t(\bar{y}, z) \wedge \exists x \leq z \ldots)$ respectively. □

Similarly to the proof in [2] concerning the theory U_2^1, it can be proved that PSPACE is the class of functions provably total in $\Sigma_1^{1,\mathsf{b}}$-NIA with $\Sigma_1^{1,b}$-graphs. The proof uses the free cut elimination theorem, after formulating the theory $\Sigma_1^{1,\mathsf{b}}$-NIA into Gentzen's sequent calculus. We start by presenting the sequent calculus formulation for $\Sigma_1^{1,\mathsf{b}}$-NIA, denoted by $\mathsf{LK}_{\mathsf{PS}}$.

Besides the initial sequents of the form $A \Rightarrow A$, with A an atomic formula, and the sequents for equality, $\mathsf{LK}_{\mathsf{PS}}$ has also the following axioms:

1) $\Rightarrow A(\bar{s})$, with A a basic axiom of $\Sigma_1^{1,\mathsf{b}}$-NIA and $\bar{s}$ terms;
2) $s \in X^t \Rightarrow s \leq t$;

3) $\Rightarrow \exists X^s \forall y \leq s\,(y \in X^s \leftrightarrow A(y))$, with A a $\Sigma_0^{1,b}$-formula where the variable X^s does not occur,

and besides the usual rules for predicate logic has the following rules for *first-order bounded quantifications:*

$$\frac{\Gamma, A(t) \longrightarrow \Delta}{\Gamma, t \leq s, \forall x \leq s A(x) \longrightarrow \Delta}\ \forall_{\leq} : l \qquad \frac{\Gamma, b \leq t \longrightarrow \Delta, A(b)}{\Gamma \longrightarrow \Delta, \forall x \leq t A(x)}\ \forall_{\leq} : r$$

$$\frac{\Gamma, b \leq t, A(b) \longrightarrow \Delta}{\Gamma, \exists x \leq t A(x) \longrightarrow \Delta}\ \exists_{\leq} : l \qquad \frac{\Gamma \longrightarrow \Delta, A(t)}{\Gamma, t \leq s \longrightarrow \Delta, \exists x \leq s A(x)}\ \exists_{\leq} : r$$

where b is an eigenvariable (i.e. it is not free in either Γ, Δ or t);
has the following rules for *second-order bounded quantifications:*

$$\frac{\Gamma, A(F^t) \longrightarrow \Delta}{\Gamma, \forall X^t A(X^t) \longrightarrow \Delta}\ \forall_{2^{nd}} : l \qquad \frac{\Gamma \longrightarrow \Delta, A(C^t)}{\Gamma \longrightarrow \Delta, \forall X^t A(X^t)}\ \forall_{2^{nd}} : r$$

$$\frac{\Gamma, A(C^t) \longrightarrow \Delta}{\Gamma, \exists X^t A(X^t) \longrightarrow \Delta}\ \exists_{2^{nd}} : l \qquad \frac{\Gamma \longrightarrow \Delta, A(F^t)}{\Gamma \longrightarrow \Delta, \exists X^t A(X^t)}\ \exists_{2^{nd}} : r$$

with F^t a second-order variable and C^t a second-order eigenvariable;
has the following *induction rule:*

$$\frac{\Gamma, A(x) \longrightarrow \Delta, A(x0) \qquad \Gamma, A(x) \longrightarrow \Delta, A(x1)}{\Gamma, A(\epsilon) \longrightarrow \Delta, A(s)}$$

with A a $\Sigma_1^{1,b}$-formula (possibly with parameters), s a term, and x an eigenvariable;
and has the following *replacement rule:*

$$\frac{\Gamma, x \leq t \longrightarrow \Delta, \exists X^q A(x, X^q)}{\Gamma \longrightarrow \Delta, \exists Y^r \forall x \leq t \bar{A}(x, Y^r)}$$

where x is an eigenvariable, A is a $\Sigma_0^{1,b}$-formula, and r and $\bar{A}$ are as in the replacement scheme.[5]

The point of this sequent calculus formulation of $\Sigma_1^{1,b}$-NIA is that, if $\mathsf{LK_{PS}} \vdash \Gamma \longrightarrow \Delta$, with Γ and Δ formed by $\Sigma_1^{1,b}$-formulas, then (by an easy application of the free cut elimination theorem) there is a $\mathsf{LK_{PS}}$-proof of $\Gamma \longrightarrow \Delta$ in which every formula appearing in the proof is a $\Sigma_1^{1,b}$-formula. Note that it is the replacement scheme that allows us to have a sequent calculus proof just with $\Sigma_1^{1,b}$-formulas instead of extended $\Sigma_1^{1,b}$-formulas.

The fundamental theorem to achieve our goal – characterize the provably total functions of $\Sigma_1^{1,b}$-NIA – is the technical result (Theorem 3.4, below), which asserts that there are extended $\Sigma_1^{1,b}$-formulas, extended $\Pi_1^{1,b}$-formulas and functionals related with polyspace which witness the second-order existential quantifications in the consequents of the sequents in the $\mathsf{LK_{PS}}$-proof above.

[5] A similar replacement rule appears in [1] in a reformulated version of the sequent calculus for U_2^1.

First some considerations about second-order computations. We denote by $\mathsf{PSPACE}^{\Sigma_0^{1,b}}$ the class of functionals computable by a deterministic Turing Machine in polynomial space with oracle $\Sigma_0^{1,b}$.[6]

Proposition 3.3. *The functionals in the class* $\mathsf{PSPACE}^{\Sigma_0^{1,b}}$ *which are functions, are exactly the polyspace computable functions.*

Proof. If $f \in \mathsf{PSPACE}$ it is immediate that $f \in \mathsf{PSPACE}^{\Sigma_0^{1,b}}$. Just notice that we can consider that the second-order tapes stay empty and the oracle is not called to intervene in the computation. For the other inclusion, let f be a function in $\mathsf{PSPACE}^{\Sigma_0^{1,b}}$. So, there is M_1 a deterministic polyspace Turing machine with a $\Sigma_0^{1,b}$ oracle, say $Q(x, X^t)$, such that when the input is $\bar{w}$, the output is $f(\bar{w})$. Let M_2 be a deterministic Turing machine without second-order tapes (and without oracle), which we initialize with $\bar{w}$ and that executes the same steps as M_1, except that, when M_1 writes symbols in the first-order oracle tape, M_2 writes them in a working tape and when M_1 writes symbols in the second-order oracle tape, M_2 does nothing. Let us see what to do when M_1 asks a question to the oracle for the first time. If $Q(x, X^t)$ is a first-order formula, since it is bounded, it is decidable in polyspace, thus in M_2 we just have to introduce the steps that decide $Q(x)$. If X^t occurs in $Q(x, X^t)$ suppose, for a simple example[7], that $Q(x, X^t)$ is of the form $\exists z_1 \leq q_1(x) \forall z_2 \leq q_2(x)\, r(x, z_1, z_2) \in X^t$. Let us argue that there is a polytime alternating Turing machine that with input x decides $Q(x, X^t)$. Since it is well-known that: (1) PSPACE is equivalent to polytime in alternating Turing machines, (2) in these machines there are existential and universal states and (3) q_1 and q_2 can be computed in PSPACE; if we manage to decide $r \in X^t$ in PSPACE there is an alternating Turing machine that decides $Q(x, X^t)$ is polytime. From x, z_1 and z_2 we can compute r in PSPACE. To decide $r \in X^t$, we can think in a machine initialized with $\bar{w}$, which acts like M_1 but when M_1 writes in the oracle first-order tape it writes in a working tape and when M_1 writes an element in the oracle second-order tape, the new machine writes in binary notation the number of elements already written in the oracle second-order tape by M_1. As in [15], we assume that the pointers in the oracle tapes just move right and exactly a cell at a time. This means that, when the counting number (in the new machine) is r, the next element to be written in the oracle second-order tape by M_1 informs us if r belongs to X^t or nor. Read it. If it is 0 then $r \notin X^t$, if it is 1 then $r \in X^t$. Note that we are assuming that X^t is written in the oracle tape in the fixed way described in footnote 6. Therefore, the decision of $r \in X^t$ is done in PSPACE. Since we saw that $Q(x, X^t)$ can be decided from x using a polytime

[6] The idea is that the machine allows first and second-order inputs, returns first and second-order outputs and allows questions to the oracle involving first and second-order parameters. A bounded set is written in a (input, output or oracle) second-order machine tape in the following way: in the j^{th} cell of the tape put 1 if the binary word for j belongs to the set, put 0 otherwise. If the first-order input tapes have length less than or equal to n and the second-order input tapes have length less than or equal to $2^{p(n)}$, the computation occurs in polynomial space if the number of tape cells used (in the working tapes) is less than or equal to n^k, with $k \in \mathbb{N}$. We say that the machine computes a *function* if it has no inputs nor outputs of second-order (having the possibility of having second-order parameters in the $\Sigma_0^{1,b}$ questions to the oracle). For a full description of this kind of Turing machines see [15], page 16.

[7] The example with bounded first-order existential and universal quantifications and X^t illustrates what happens when the $\Sigma_0^{1,b}$-oracle is call to intervene.

alternating Turing machine, it can be decided in PSPACE. In M_2 just introduce the steps to decide $Q(x, X^t)$. We assume that after every oracle query, M_1 clears the oracle tape and M_2 saves the state of M_1. In the next call to the oracle, when simulating to see if $r \in X^t$, we initialize the machine with this state rather than with $\bar{w}$. Repeating the process everytime M_1 consults the oracle, we have that M_2 computes f in polynomial space. Thus, $f \in$ PSPACE. □

In order to simplify notation, in the next theorems we omit the bounding term in the second-order variables and we abbreviate $X_1, ..., X_n$, with $n \in \mathbb{N}$, by $\bar{X}$ and $\exists X_1, ..., \exists X_n$ (respectively $\forall X_1, ..., \forall X_n$) by $\exists \bar{X}$ (respectively $\forall \bar{X}$).

Theorem 3.4. *Suppose* $\mathsf{LK_{PS}} \vdash \Gamma \longrightarrow \Delta$*, where* Γ *and* Δ *are formed by* $\Sigma_1^{1,b}$*-formulas. Consider* $\Gamma := \exists \bar{X}\varphi_1(\bar{x}, \bar{F}, \bar{X}), ..., \exists \bar{X}\varphi_n(\bar{x}, \bar{F}, \bar{X})$ *and* $\Delta := \exists \bar{Y}\psi_1(\bar{x}, \bar{F}, \bar{Y}), ..., \exists \bar{Y}\psi_m(\bar{x}, \bar{F}, \bar{Y})$ *where* φ_i *and* ψ_i *are* $\Sigma_0^{1,b}$*-formulas,* $\bar{Y} = Y_1, ..., Y_k$ *and* $\varphi_1, ..., \varphi_n$, $\psi_1, ..., \psi_m$ *have different components of* $\bar{X}$, $\bar{Y}$ *respectively.*

Consider $\varphi(\bar{x}, \bar{F}, \bar{X}) := \bigwedge_{j=1}^{n} \varphi_j$ *and* $\psi(\bar{x}, \bar{F}, \bar{Y}) := \bigvee_{i=1}^{m} \psi_i$ *and denote by* $\theta(\bar{x}, \bar{F}, \bar{X}, \bar{Y})$ *the formula* $\varphi(\bar{x}, \bar{F}, \bar{X}) \to \psi(\bar{x}, \bar{F}, \bar{Y})$.

Then, there are terms $t_i(\bar{x})$*, extended* $\Sigma_1^{1,b}$*-formulas* $M_i^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X})$*, extended* $\Pi_1^{1,b}$*-formulas* $M_i^{\Pi}(w, \bar{x}, \bar{F}, \bar{X})$ *and functionals* $\phi_i(\bar{x}, \bar{F}, \bar{X})$ *in* $\mathsf{PSPACE}^{\Sigma_0^{1,b}}$ $(1 \leq i \leq k)$*, such that:*
$\Sigma_1^{1,b}$-NIA $\vdash \forall \bar{x} \forall \bar{F} \forall \bar{X} \theta(\bar{x}, \bar{F}, \bar{X}, \{w \leq t_1(\bar{x}) : M_1^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X})\}, ..., \{w \leq t_k(\bar{x}) : M_k^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X})\})$*;*
$\Sigma_1^{1,b}$-NIA $\vdash \forall \bar{x} \forall \bar{F} \forall \bar{X} \forall w \leq t_i(\bar{x})(M_i^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X}) \leftrightarrow M_i^{\Pi}(w, \bar{x}, \bar{F}, \bar{X})), (1 \leq i \leq k)$ *and* $\phi_i(\bar{x}, \bar{F}, \bar{X}) = \{w \leq t_i(\bar{x}) : M_i^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X})\}$*,* $(1 \leq i \leq k)$*.*

In the previous result, we denote by $\theta(\bar{x}, \bar{F}, \bar{X}, \{w \leq t_1(\bar{x}) : M_1^{\Sigma}(w, \bar{x}, \bar{F}, \bar{X})\}, ...)$ the formula $\theta(\bar{x}, \bar{F}, \bar{X}, G, ...)$ where the occurrences of $s \in G$ are replaced by $s \leq t_1(\bar{x}) \wedge M_1^{\Sigma}(s, \bar{x}, \bar{F}, \bar{X})$.

Proof. Let $\mathcal{P}$ be a $\mathsf{LK_{PS}}$-proof of $\Gamma \longrightarrow \Delta$. By the free cut elimination theorem we can suppose that $\mathcal{P}$ has just $\Sigma_1^{1,b}$-formulas. It is possible to prove, by induction on the number of lines in $\mathcal{P}$, that for every sequent $\Pi \longrightarrow \Lambda$ in $\mathcal{P}$, there exist the terms, the formulas and the functionals described in the theorem.

We illustrate the strategy with the cases of comprehension and replacement. For a detailed proof see [15], pages 50–55.

Comprehension. If $\Pi \longrightarrow \Lambda$ is an initial sequent of the form $\longrightarrow \exists Y^s \forall y \leq s(y \in Y^s \leftrightarrow A(y, \bar{x}, \bar{F}))$, with A a $\Sigma_0^{1,b}$-formula, just define $M_1^{\Sigma}(w, \bar{x}, \bar{F})$ and $M_1^{\Pi}(w, \bar{x}, \bar{F})$ as being $A(w, \bar{x}, \bar{F})$. Note that a $\Sigma_0^{1,b}$-formula is, in particular, an extended $\Sigma_1^{1,b}$-formula and an extended $\Pi_1^{1,b}$-formula. Take $t_1 :\equiv s$ and $\phi_1(\bar{x}, \bar{F}) :\equiv \{w \leq s : A(w, \bar{x}, \bar{F})\}$.

Replacement. Suppose $\Pi \longrightarrow \Lambda$ is $\Gamma' \longrightarrow \Delta', \exists G \forall x \leq t\bar{\varphi}(x, G)$, obtained from the sequent $\Gamma', z \leq t \longrightarrow \Delta', \exists Y \varphi(z, Y)$, with φ a $\Sigma_0^{1,b}$-formula (the rule is only applied to formulas of this complexity). For simplicity of notation we assume that Δ' has only the single formula $\exists Z \psi_1(\bar{x}, \bar{X}, Z)$, with ψ_1 a $\Sigma_0^{1,b}$-formula. By induction hypothesis for the last sequent there exist $(M')_i^{\Sigma}(w, z, \bar{x}, \bar{X})$, $(M')_i^{\Pi}(w, z, \bar{x}, \bar{X})$, $t_i'(z, \bar{x})$ and $\phi_i'(z, \bar{x}, \bar{X})$, for $i \in \{1, 2\}$, in the desired conditions. For the sequent $\Gamma' \longrightarrow \exists Z \psi_1(\bar{x}, \bar{X}, Z), \exists G \forall x \leq t\bar{\varphi}(x, G)$ define $M_1^{\Sigma}(w, \bar{x}, \bar{X})$ as being

$\exists z \leq t(\neg\varphi(z, \{w \leq t'_2(z,\bar{x}) : (M')^{\Delta}_2(w,z,\bar{x},\bar{X})\}) \wedge \forall z' < z\varphi(z', \{w \leq t'_2(z',\bar{x}) : (M')^{\Delta}_2(w,z',\bar{x},\bar{X})\}) \wedge w \leq t'_1(z,\bar{x}) \wedge (M')^{\Sigma}_1(w,z,\bar{x},\bar{X}))$[8];
$t_1(\bar{x})$ as being $t'_1(t(\bar{x}),\bar{x})$; $\phi_1(\bar{x},\bar{X})$ as being $\emptyset$ if $\forall z \leq t\varphi(z,\phi'_2(z,\bar{x},\bar{X}))$ or $\phi_1(\bar{x},\bar{X})$ as being $\phi'_1(\mu z \leq t\neg\varphi(z,\phi'_2(z,\bar{x},\bar{X})),\bar{x},\bar{X})$ if $\exists z \leq t\neg\varphi(z,\phi'_2(z,\bar{x},\bar{X}))$[9]; define $M^{\Sigma}_2(w,\bar{x},\bar{X})$ as being $\exists z \leq t\exists s \leq t'_2(z,\bar{x})((M')^{\Sigma}_2(s,z,\bar{x},\bar{X}) \wedge w =< z, s >)$; $t_2(\bar{x}) := (t(\bar{x})^\frown t'_2(t(\bar{x}),\bar{x}))1 \times 11$ and $\phi_2(\bar{x},\bar{X}) :\equiv \{< z, s >: z \leq t \wedge s \leq t'_2 \wedge s \in \phi'_2(z,\bar{x},\bar{X})\}$. And define $M^{\Pi}_1(w,\bar{x},\bar{X})$ and $M^{\Pi}_2(w,\bar{x},\bar{X})$ like the above Σ-versions with the expected changes. □

From the previous result we get the following theorem:

Theorem 3.5. *If* $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\exists yA(\bar{x},y)$*, with* A *a* $\Sigma^{1,b}_1$*-formula, then there exists a function* $f \in$ PSPACE *such that, for all* $\bar{\sigma} \in 2^{<\omega}$*, we have* $A(\bar{\sigma}, f(\bar{\sigma}))$*.*

Moreover, there exists θ*, a formula that is equivalent (in* $\Sigma^{1,b}_1$-NIA*) to both an extended* $\Sigma^{1,b}_1$*-formula and an extended* $\Pi^{1,b}_1$*-formula, and a term* $t(\bar{x})$*, such that:*

1) $f(\bar{\sigma}) = \tau \leftrightarrow \theta(\bar{\sigma},\tau)$*,*

2) $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\forall y(\theta(\bar{x},y) \to A(\bar{x},y))$*,*

3) $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\exists y \leq t\theta(\bar{x},y)$*,*

4) $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\exists^1 y\theta(\bar{x},y)$*.*

Proof. Since $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\exists yA(\bar{x},y)$, with A a $\Sigma^{1,b}_1$-formula it is possible to prove that there exists $t'(\bar{x})$ such that $\Sigma^{1,b}_1$-NIA $\vdash \forall\bar{x}\exists y \leq t'(\bar{x})A(\bar{x},y)$. Thus $\mathsf{LK}_{\mathsf{PS}}$ deduce the sequent $\longrightarrow \exists y \leq t'(\bar{x})A(\bar{x},y)$. Being A a $\Sigma^{1,b}_1$-formula, suppose it is $\exists U_1 ... \exists U_k \tilde{A}$, with $\tilde{A}$ a $\Sigma^{1,b}_0$-formula. So there exists a $\mathsf{LK}_{\mathsf{PS}}$-proof of $\longrightarrow \exists U_1 ... \exists U_k \exists y \leq t'\tilde{A}$. The result follows applying Theorem 3.4 and defining $f(\bar{\sigma})$ as being $\mu y \leq t'(\bar{\sigma})\tilde{A}(\bar{\sigma}, y, \phi_1(\bar{\sigma}), \ldots, \phi_k(\bar{\sigma}))$, $\theta(\bar{x},y) :\leftrightarrow \tilde{A}(\bar{x},y,\{w \leq t_1(\bar{x}) : M^{\Delta}_1(w,\bar{x})\}, ..., \{w \leq t_k(\bar{x}) : M^{\Delta}_k(w,\bar{x})\}) \wedge \forall y' <_l y\neg\tilde{A}(\bar{x},y',\{w \leq t_1(\bar{x}) : M^{\Delta}_1(w,\bar{x})\}, ..., \{w \leq t_k(\bar{x}) : M^{\Delta}_k(w,\bar{x})\})$ and $t(\bar{x}) :\equiv t'(\bar{x})$, where the ϕ_i, t_i and M^{Δ}_i come from Theorem 3.4. To confirm that f is a polyspace function use Proposition 3.3. □

So, the provably total functions in $\Sigma^{1,b}_1$-NIA, with $\Sigma^{1,b}_1$-graphs, are exactly the functions of PSPACE.

4. Adding bounded collection

Having in view setting up a system of weak analysis connected with polyspace computability, in this (intermediate) section we enrich the theory $\Sigma^{1,b}_1$-NIA with a bounded collection scheme. The idea is that the new scheme does not increase the computational power of the enriched theory and allows (as shown in the next section) the inclusion of

[8] M^{Δ}_i is either M^{Σ}_i or M^{Π}_i according to our conveniences (e.g. to have the formulas in the right classes).

[9] By μzA we mean the least z satisfying A. It is possible to prove that the minimization scheme for formulas that are equivalent to both an extended $\Sigma^{1,b}_1$-formula and an extended $\Pi^{1,b}_1$-formula holds in $\Sigma^{1,b}_1$-NIA. See [15], page 34.

recursive comprehension (an essential scheme to the formalization of analysis). The inspiration comes again from the polytime setting (see [10]) where the principle of bounded collection

$\mathsf{B}\Sigma^{\mathsf{b}}_{\infty}$: $\forall x \leq t \exists y A(x, y) \to \exists z \forall x \leq t \exists y \leq z A(x, y)$,

where A is a formula of $\mathcal{L}$ with all quantifications bounded and z is a new variable, is appended to Σ^{b}_1-NIA. See also [3, 11].

In the current context of polyspace computability, we choose a form of bounded collection slightly strengthened. See the comments in the end of the paper.

To prove, in this section, that the system $\Sigma^{1,\mathsf{b}}_1$-NIA enriched with (the stronger form of) bounded collection still characterizes polyspace computability, we rely on a conservation result over $\Sigma^{1,\mathsf{b}}_1$-NIA and on the cut-elimination technique.

The *bounded formulas* of $\mathcal{L}^b_2$, also known as the $\Sigma^{1,b}_{\infty}$-formulas, consist of the smallest class of formulas containing the atomic formulas and closed under Boolean connectives, first-order bounded quantifications and second-order (bounded) quantifications.

We are going to enrich $\Sigma^{1,\mathsf{b}}_1$-NIA with the following principle of *bounded collection*.

$\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$: $\forall X^t \exists y A(y, X^t) \to \exists z \forall X^t \exists y \leq z A(y, X^t)$,

where A is a bounded formula (possibly with parameters) and z is a new variable.

Note that the principle $\mathsf{B}\Sigma^{\mathsf{b}}_{\infty}$ can be derived in the system $\Sigma^{1,\mathsf{b}}_1$-NIA + $\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$.

Theorem 4.1. *The theory $\Sigma^{1,\mathsf{b}}_1$-NIA + $\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$ is conservative over the theory $\Sigma^{1,\mathsf{b}}_1$-NIA with respect to formulas of the form $\forall \bar{x} \exists y A(\bar{x}, y)$, with A a $\Sigma^{1,b}_{\infty}$-formula.*

Proof. The theory $\Sigma^{1,\mathsf{b}}_1$-NIA + $\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$ can be formulated in the sequent calculus described above, $\mathsf{LK_{PS}}$, together with the following rule for bounded collection ($\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$-rule):

$$\frac{\Gamma \longrightarrow \Delta, \exists y A(y, C^t)}{\Gamma \longrightarrow \Delta, \exists z \forall X^t \exists y \leq z A(y, X^t)}$$

where A is a $\Sigma^{1,b}_{\infty}$-formula (possibly with other free variables), C^t is a second-order eigenvariable, and y does not occur in the term t.

Suppose that $\Sigma^{1,\mathsf{b}}_1$-NIA + $\mathsf{B}^1\Sigma^{1,\mathsf{b}}_{\infty}$ $\vdash$ $\forall \bar{x} \exists y A(\bar{x}, y)$, with A a $\Sigma^{1,b}_{\infty}$-formula. Thus, in the sequent calculus above, there is a proof of $\longrightarrow \forall \bar{x} \exists y A(\bar{x}, y)$ and so, a proof of $\longrightarrow \exists y A(\bar{x}, y)$. The free cut elimination theorem ensures the existence of a proof $\mathcal{P}$ of $\longrightarrow \exists y A(\bar{x}, y)$ without free cuts. As a consequence, all formulas occurring in the sequents in $\mathcal{P}$ are of the form $\exists y B(y, \bar{x}, \bar{X}^p)$, with B a $\Sigma^{1,b}_{\infty}$-formula (the existential quantifiers "$\exists y$" may be absent in which case we have a bounded formula).

Let $\Gamma \longrightarrow \Delta$ be a sequent in $\mathcal{P}$, where Γ is $\exists x_1 B_1(x_1, \bar{x}, \bar{X}^{p(\bar{x})}), \ldots, \exists x_n B_n(x_n, \bar{x}, \bar{X}^{p(\bar{x})})$ and Δ is $\exists y_1 C_1(y_1, \bar{x}, \bar{X}^{p(\bar{x})}), \ldots, \exists y_k C_k(y_k, \bar{x}, \bar{X}^{p(\bar{x})})$, where $B_1, \ldots, B_n, C_1, \ldots, C_k$ are $\Sigma^{1,b}_{\infty}$-formulas. To ease notation, we have the same term $p(\bar{x})$ for every formula (this can be assumed without loss of generality) and we are also only displaying the variables $\bar{x}$ in the term $p(\bar{x})$.

Let us prove, by induction on the number of lines of $\mathcal{P}$, that from bounds of the antecedents we can get (in $\Sigma^{1,\mathsf{b}}_1$-NIA) bounds for the consequents in the following way:

$$\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v}),$$

where $\Gamma^{\leq u}$ abbreviates $\exists x_1 \leq uB_1(x_1, \bar{x}, \bar{X}^{p(\bar{x})}) \wedge \ldots \wedge \exists x_n \leq uB_n(x_n, \bar{x}, \bar{X}^{p(\bar{x})})$ and $\Delta^{\leq v}$ abbreviates $\exists y_1 \leq vC_1(y_1, \bar{x}, \bar{X}^{p(\bar{x})}) \vee \ldots \vee \exists y_k \leq vC_k(y_k, \bar{x}, \bar{X}^{p(\bar{x})})$. Note that applying this result to the last sequent of $\mathcal{P}$, we conclude that $\mathsf{\Sigma_1^{1,b}\text{-}NIA}$ proves $\forall u \exists v \forall \bar{x} \leq u \exists y \leq vA(\bar{x}, y)$. Thus $\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall \bar{x} \exists y A(\bar{x}, y)$.

The proof above (by induction on the number of lines of $\mathcal{P}$) has some trivial cases. Here we illustrate two situations: when the sequent is obtained from the cut-rule and when it is obtained from the $\mathsf{B^1\Sigma_\infty^{1,b}}$-rule. For a complete proof of the result, considering all the cases, see [15], pages 58–60.

Cut-rule.

$$\frac{\Gamma \longrightarrow \Delta, A \qquad A, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

If A is a $\Sigma_\infty^{1,b}$-formula the result is immediate taking $v :\equiv v_1 \hat{\ } v_2$, where v_1 and v_2 the bounds that exist by induction hypothesis.

Suppose that A is of the form $\exists z D(z, \bar{x}, X^{p(\bar{x})})$ with D a $\Sigma_\infty^{1,b}$-formula. By induction hypothesis we have:

1) $\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v} \vee \exists z \leq vD(z, \bar{x}, \bar{X}^{p(\bar{x})}))$

2) $\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\exists z \leq uD(z, \bar{x}, \bar{X}^{p(\bar{x})}) \wedge \Gamma^{\leq u} \to \Delta^{\leq v})$.

Our goal is to prove that $\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v})$.

Fix u. By 1) there is v_1 such that $\forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v_1} \vee \exists z \leq v_1 D(z, \bar{x}, \bar{X}^{p(\bar{x})}))$. Assume that $u \leq v_1$ (if not just replace v_1 by $v_1 \hat{\ } u$ that the assertion above remains valid). By 2), there is v_2 such that $\forall \bar{x} \leq v_1 \forall \bar{X}^{p(\bar{x})}(\exists z \leq v_1 D(z, \bar{x}, \bar{X}^{p(\bar{x})}) \wedge \Gamma^{\leq v_1} \to \Delta^{\leq v_2})$. We get $\forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\exists z \leq v_1 D(z, \bar{x}, \bar{X}^{p(\bar{x})}) \wedge \Gamma^{\leq u} \to \Delta^{\leq v_2})$, using the fact that if $u \leq u'$ then $\Gamma^{\leq u} \to \Gamma^{\leq u'}$. Let $v :\equiv v_1 \hat{\ } v_2$. We conclude that $\forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v})$.

$\mathsf{B^1\Sigma_\infty^{1,b}}$*-rule.*

$$\frac{\Gamma \longrightarrow \Delta, \exists y A(y, C^t)}{\Gamma \longrightarrow \Delta, \exists z \forall Y^t \exists y \leq zA(y, Y^t)}$$

By induction hypothesis we know that

$$\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})} \forall Y^{t(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v} \vee \exists y \leq vA(y, Y^{t(\bar{x})})).$$

To prove, as we want, that

$$\mathsf{\Sigma_1^{1,b}\text{-}NIA} \vdash \forall u \exists v \forall \bar{x} \leq u \forall \bar{X}^{p(\bar{x})}(\Gamma^{\leq u} \to \Delta^{\leq v} \vee \exists z \leq v \forall Y^{t(\bar{x})} \exists y \leq zA(y, Y^{t(\bar{x})})),$$

consider an arbitrary u, take v as the element that exists by induction hypothesis, and let $z :\equiv v$. □

From the above result, noticing that $\Sigma_1^{1,b}$-formulas are, in particupar, $\Sigma_\infty^{1,b}$-formulas, immediately we conclude that the provably total functions of $\mathsf{\Sigma_1^{1,b}\text{-}NIA} + \mathsf{B^1\Sigma_\infty^{1,b}}$ with $\Sigma_1^{1,b}$-graphs are the polyspace computable functions.

5. A weak analysis theory for polyspace computability

The aim of this section is to propose a weak theory of analysis, BTPSA, still characterizing polyspace computability. In [10], an homologous theory, BTFA, is presented, not for PSPACE complexity, but in the context of feasible (polytime) computability. The strategy in the polytime setting will guide us towards our goal.

The theory BTPSA, as we are going to see, includes the previous system $\Sigma_1^{1,b}$-NIA + $B^1\Sigma_\infty^{1,b}$ and is stated in a language that permits variables ranging over infinite sets. Let $\mathcal{L}_2$ be a second-order language with equality which differs from $\mathcal{L}_2^b$ only by the presence of second-order (unbounded) variables, denoted by $X, Y, Z \ldots$, instead of the previous second-order "bounded" variables $X^t, Y^q, \ldots$ The class of formulas of $\mathcal{L}_2$ is the smallest class of expressions containing the atomic formulas $t_1 \subseteq t_2, t_1 = t_2, t_1 \in X$, and closed under the Boolean operations, the first-order quantifications $\forall x, \exists x$ and the second-order quantifications $\forall X, \exists X$. Since arguments via sequent calculus are not going to be used, in BTPSA we do not treat $\forall x \leq t$ and $\exists x \leq t$ as primitive quantifiers but mere abbreviations of $\forall x(x \leq t \to \ldots)$ and $\exists x(x \leq t \wedge \ldots)$ respectively. The definitions of $\Sigma_0^{1,b}$ (respectively $\Sigma_1^{1,b}$, $\Pi_1^{1,b}$, $\Sigma_\infty^{1,b}$)-formulas in $\mathcal{L}_2$ (and its extended versions) are given by obvious modifications of the homologous definitions in $\mathcal{L}_2^b$, namely replacing $\forall X^t$, $\exists X^t$ by the second-order quantifications $\forall X \leq t$, $\exists X \leq t$ where $X \leq t$ abbreviates $\forall z(z \in X \to z \leq t)$.

A structure for $\mathcal{L}_2$ has domain (M, S), with the first-order variables taking values in M and the second-order variables varying over S, a given subset of $\mathcal{P}(M)$. The standard model is $(2^{<\omega}, \mathcal{P}(2^{<\omega}))$. Note that although we work in a second-order language, our logic is of first-order kind (first-order logic in a two-sorted language): our semantics only specifies S to be a subset of $\mathcal{P}(M)$, not necessarily all of $\mathcal{P}(M)$.

Consider the following axiom, known as the *recursive comprehension scheme*:

$$\forall x(\exists y A(x, y) \leftrightarrow \forall y B(x, y)) \to \exists X \forall x(x \in X \leftrightarrow \exists y A(x, y)),$$

with A a $\Sigma_1^{1,b}$-formula and B a $\Pi_1^{1,b}$-formula, possibly with other free variables.
In the standard model, this scheme ensures that all recursive sets exist. Although it may seem that adding this scheme to our weak (subexponential) theory will increase its computational power, we will prove that this is not the case. Note that the existence of a set is guaranteed only in the case the theory has enough resources to prove the equivalence in the antecedent of the scheme.

Definition 5.1. BTPSA (acronym for *Base Theory for Polynomial Space Analysis*) is the second-order theory, in the language $\mathcal{L}_2$, with the following axioms: basic axioms[10], induction on notation for extended $\Sigma_1^{1,b}$-formulas, the bounded collection scheme $B^1\Sigma_\infty^{1,b}$ and the recursive comprehension scheme mentioned above.

Intuitively $\mathcal{L}_2^b \subseteq \mathcal{L}_2$, in the sense that every expression in $\mathcal{L}_2^b$ can be formulated in $\mathcal{L}_2$. Also, it can be seen that every model of BTPSA satisfies the axioms of $\Sigma_1^{1,b}$-NIA + $B^1\Sigma_\infty^{1,b}$. To achieve our goal (i.e. to prove that BTPSA characterizes polyspace computability) we will do the inverse, i.e., to get models of BTPSA from models of $\Sigma_1^{1,b}$-NIA + $B^1\Sigma_\infty^{1,b}$. Notice

[10]The 14 basic axioms of Definition 2.3.

the model-theoretic strategy instead of the proof-theoretic techniques by cut-elimination in previous conservation results.

Lemma 5.2. *Let $\mathcal{M}$ be a model of the theory $\Sigma^{1,b}_1$-NIA + $B^1\Sigma^{1,b}_\infty$ with domain (M, S_b). Then there is $S \subseteq \mathcal{P}(M)$ such that $\mathcal{M}^*$, with domain (M, S), is a model of* BTPSA *and $S_b = \{X^a : X \in S \wedge a \in M\}$, where X^a collects the elements of X with length less than or equal to a.*

Proof. In order to get $\mathcal{M}^*$ from $\mathcal{M}$ the idea is to "close" $\mathcal{M}$ for recursive comprehension. Let S be formed by the subsets $X \subseteq M$ for which there is a $\Sigma^{1,b}_1$-formula A, a $\Pi^{1,b}_1$-formula B and elements $\bar{a}, \bar{b}$ in M and $\bar{A}^p, \bar{B}^u$ in S_b such that $X = \{x \in M : \mathcal{M} \vDash \exists y\, A(x, y, \bar{a}, \bar{A}^{p(x,y,\bar{a})})\} = \{x \in M : \mathcal{M} \vDash \forall y\, B(x, y, \bar{b}, \bar{B}^{u(x,y,\bar{b})})\}$. The proof that $S_b \subseteq \{X^a : X \in S \wedge a \in M\}$ follows immediately because whenever $C^c \in S_b$, we have $C^c \in S$. For the other inclusion consider $C \in S$ and $c \in M$. We want to prove that $C^c \in S_b$. By definition of S there are formulas A, B and elements $\bar{a}, \bar{b}, \bar{A}, \bar{B}$ (to simplify notation we omit the bounded term in the second-order parameters) - the ones defining C - such that $\mathcal{M} \vDash \forall x(\exists y A(x, y, \bar{a}, \bar{A}) \leftrightarrow \forall y B(x, y, \bar{b}, \bar{B}))$. We claim that there is $d \in M$ such that $\forall x \leq c(\exists y \leq d A(x, y, \bar{a}, \bar{A}) \leftrightarrow \forall y \leq d B(x, y, \bar{b}, \bar{B}))$. The existence of this d uses the fact that from $\forall x \leq c \exists y\, \theta(x, y)$, with θ a $\Sigma^{1,b}_\infty$-formula, we have $\exists d \forall x \leq c \exists y \leq d\, \theta(x, y)$, which is a consequence of $\mathcal{M}$ being a model of $B^1\Sigma^{1,b}_\infty$. Therefore, since $\Delta^{1,b}_1$-bounded comprehension is available in $\Sigma^{1,b}_1$-NIA (see the comments below Definition 3.1) and $\mathcal{M}$ is a model of $\Sigma^{1,b}_1$-NIA, we have that $F^c = \{x \leq c : \exists y \leq d A(x, y, \bar{a}, \bar{A})\}$ is an element in S_b. Since $F^c = C^c$, we have $C^c \in S_b$.

It remains to prove that $\mathcal{M}^*$ is a model of BTPSA. The case of the basic axioms is trivial, because the interpretation of the constants, the function and relation symbols is the same in $\mathcal{M}$ and $\mathcal{M}^*$ and both models have the same first-order domain. The study of the other axioms follows more or less in a straightforward manner from the following technical fact (which can be proved by induction on the complexity of A): *Given $A(\bar{u}, \bar{U})$ a $\Sigma^{1,b}_\infty$-formula, there is a term $q(\bar{u})$ with the following property: given $\bar{c}$ elements in M and $\bar{C}$ subsets in S then $\mathcal{M}^* \vDash A(\bar{c}, \bar{C}) \Leftrightarrow \mathcal{M} \vDash A(\bar{c}, \bar{C}^b)$ whenever $q(\bar{c}) \leq b$.* We illustrate the application of the fact studying the recursive comprehension scheme. Suppose that $\mathcal{M}^* \vDash \forall x(\exists y A(x, y, \bar{a}, \bar{A}) \leftrightarrow \forall y B(x, y, \bar{b}, \bar{B}))$ with A a $\Sigma^{1,b}_1$-formula and B a $\Pi^{1,b}_1$-formula. We want to prove that $\mathcal{M}^* \vDash \exists X \forall x(x \in X \leftrightarrow \exists y A(x, y, \bar{a}, \bar{A}))$. Note that the formulas $A(x, y, \bar{u}, \bar{U})$ and $B(x, y, \bar{u}, \bar{U})$ are, in particular, $\Sigma^{1,b}_\infty$-formulas. Applying the fact above to the formula A, there exists a term $q(x, y, \bar{u})$ such that given $s, r, \bar{a} \in M$ and $\bar{A} \in S$ we have (†) $\mathcal{M}^* \vDash A(s, r, \bar{a}, \bar{A}) \Leftrightarrow \mathcal{M} \vDash A(s, r, \bar{a}, \bar{A}^b)$ whenever $q(s, r, \bar{a}) \leq b$. Applying the same fact to B, there exists a term $p(x, y, \bar{u})$ such that given $s, r, \bar{b} \in M$ and $\bar{B} \in S$ we have $\mathcal{M}^* \vDash B(s, r, \bar{b}, \bar{B}) \Leftrightarrow \mathcal{M} \vDash B(s, r, \bar{b}, \bar{B}^{b'})$ whenever $p(s, r, \bar{b}) \leq b'$. Since $\mathcal{M}^* \vDash \forall x(\exists y A(x, y, \bar{a}, \bar{A}) \leftrightarrow \forall y B(x, y, \bar{b}, \bar{B}))$ we know that $\mathcal{M} \vDash \forall x(\exists y A(x, y, \bar{a}, \bar{A}^{q(x,y,\bar{a})}) \leftrightarrow \forall y B(x, y, \bar{b}, \bar{B}^{p(x,y,\bar{b})}))$. Take $X = \{x \in M : \mathcal{M} \vDash \exists y A(x, y, \bar{a}, \bar{A}^{q(x,y,\bar{a})})\} = \{x \in M : \mathcal{M} \vDash \forall y B(x, y, \bar{b}, \bar{B}^{p(x,y,\bar{b})})\}$. By the definition of S, $X \in S$. From (†) we know that $\mathcal{M}^* \vDash \exists y A(s, y, \bar{a}, \bar{A}) \Leftrightarrow \mathcal{M} \vDash \exists y A(s, y, \bar{a}, \bar{A}^{q(s,y,\bar{a})})$, so $X = \{x \in M : \mathcal{M}^* \vDash \exists y A(x, y, \bar{a}, \bar{A})\}$. Thus $\mathcal{M}^* \vDash \exists X \forall x(x \in X \leftrightarrow \exists y A(x, y, \bar{a}, \bar{A}))$. □

Theorem 5.3. *If* BTPSA $\vdash \forall \bar{x} \exists y A(\bar{x}, y)$, *with A a $\Sigma^{1,b}_\infty$-formula, then* $\Sigma^{1,b}_1$-NIA $\vdash \forall \bar{x} \exists y A(\bar{x}, y)$.

Proof. Suppose that $\Sigma_1^{1,b}$-NIA $\nvdash \forall\bar{x}\exists y A(\bar{x}, y)$, with A a $\Sigma_\infty^{1,b}$-formula. Therefore, by Theorem 4.1, we also have $\Sigma_1^{1,b}$-NIA + $\mathsf{B}^1\Sigma_\infty^{1,b} \nvdash \forall\bar{x}\exists y A(\bar{x}, y)$. By the completeness theorem, there is a model $\mathcal{M}$ of $\Sigma_1^{1,b}$-NIA + $\mathsf{B}^1\Sigma_\infty^{1,b}$, with domain (M, S_b), and $\bar{a} \in M$ such that $\mathcal{M} \models \forall y \neg A(\bar{a}, y)$. Using the previous lemma, take $S \subseteq \mathcal{P}(M)$ such that $\mathcal{M}^*$ with domain (M, S) is a model of BTPSA and $S_b = \{X^a : X \in S \wedge a \in M\}$. Clearly, $\Sigma_\infty^{1,b}$-formulas are absolute between $\mathcal{M}$ and $\mathcal{M}^*$. Therefore, we also have $\mathcal{M}^* \models \forall y \neg A(\bar{a}, y)$. Since $\mathcal{M}^*$ is a model of BTPSA, by soundness we conclude that BTPSA $\nvdash \forall\bar{x}\exists y A(\bar{x}, y)$. □

As a consequence, the provably total functions of BTPSA, with $\Sigma_1^{1,b}$-graphs, are still the polyspace computable functions. For more on BTPSA, including its interpretability in Robinson's theory Q, see [14].

Although the formalization of analysis in weak systems is out of the scope of the present paper, we just want to point to the reader that the system BTPSA is strong enough to formalize the Riemann integral, up to the fundamental theorem of calculus, for functions with a modulus of uniform continuity. See [16, 13]. Such formalization does not seem to be possible in the polytime computability setting (BTFA), unless some classes of computational complexity collapse. See [12]. Nevertheless, a non trivial amount of analysis ([6]) is already available in BTFA.

The reason we opted to equip BTPSA with a bounded collection scheme stronger than the one available in BTFA is twofold. First, as we saw, (although stronger) it is still conservative over $\Sigma_1^{1,b}$-NIA. Secondly, it is known that to prove that the following *strict* Π_1^1-*reflection* principle (a strong form of weak König's lemma):

$$\forall X \exists x A(X, x) \to \exists z \forall X \exists x \leq z A(X, x),$$

where A is a $\Sigma_\infty^{1,b}$-formula, is first-order conservative over BTPSA, we need the stronger bounded collection scheme $\mathsf{B}^1\Sigma_\infty^{1,b}$. See [5, 16]. And in BTPSA enriched with the previous reflection principle, the Riemann integral is available for continuous functions (no need for the modulus of uniform continuity assumption).

Acknowledgments

The second author is very grateful to Fernando Ferreira for the help and guidance in achieving the results of this paper. Many thanks also to the anonymous referee for the detailed revision and comments that so much improved the final version of this paper.

References

[1] A. Beckmann and S. R. Buss. Improved witnessing and local improvement principles for second-order bounded arithmetic. To appear in *ACM Transactions on Computational Logic*.

[2] S. R. Buss. *Bounded arithmetic*. PhD thesis, Princeton University, Princeton, New Jersey, 1985. A revision of this thesis was published by Bibliopolis (Naples) in 1986.

[3] S. R. Buss. A conservation result concerning bounded theories and the collection scheme. *Proceedings of the American Mathematical Society*, 100:109–116, 1987.

[4] P. Clote and G. Takeuti. First order bounded arithmetic and small boolean circuit complexity classes. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 154–218, Birkhäuser, Boston, 1995.

[5] A. M. Fernandes. *Investigações em sistemas de análise exequível* (in Portuguese). PhD thesis, Universidade de Lisboa, 2001.

[6] A. M. Fernandes and F. Ferreira. Groundwork for weak analysis. *The Journal of Symbolic Logic*, 67(2):557–578, 2002.

[7] A. M. Fernandes, F. Ferreira, and G. Ferreira. Techniques in weak analysis for conservation results. *New Studies in Weak Arithmetics*, P. Cégielski, Ch. Cornaros and C. Dimitracopoulos (eds.), CSLI Lecture Notes Number 211, pp. 115-147, 2013.

[8] F. Ferreira. *Polynomial time computable arithmetic and conservative extensions*. PhD thesis, Pennsylvania State University, USA, 1988.

[9] F. Ferreira. Polynomial time computable arithmetic. *Contemporary Mathematics*, 106:137–156, 1990.

[10] F. Ferreira. A feasible theory for analysis. *The Journal of Symbolic Logic*, 59(3):1001–1011, 1994.

[11] F. Ferreira. A note on a result of Buss concerning bounded theories and the collection scheme. *Portugaliae Mathematica*, 52:331–336, 1995.

[12] F. Ferreira and G. Ferreira. Counting as integration in feasible analysis. *Mathematical Logic Quaterly*, 52(3):315 – 320, 2006.

[13] F. Ferreira and G. Ferreira. The Riemann integral in weak systems of analysis. *Journal of Universal Computer Science*, 14(6):908–937, 2008.

[14] F. Ferreira and G. Ferreira. Interpretability in Robinson's Q. *The Bulletin of Symbolic Logic*, 19(3):289–317, 2013.

[15] G. Ferreira. *Aritmética Computável em Espaço Polinomial* (in Portuguese). Master thesis, Universidade de Lisboa, 2001. http://www.eecs.qmul.ac.uk/∼gilda/

[16] G. Ferreira. *Sistemas de Análise Fraca para a Integração* (in Portuguese). PhD thesis, Universidade de Lisboa, 2006.

[17] G. Ferreira and I. Oitavem. An interpretation of S^1_2 in Σ^b_1-NIA. *Portugaliae Mathematica*, 63(4):427–450, 2006.

[18] J. Johannsen and C. Pollett. On proofs about threshould circuits and counting hierarchies (extended abstract). In *Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 444–452, 1998.

[19] L. Kołodziejczyk, P. Nguyen, and N. Thapen. The provably total NP search problems of weak second order bounded arithmetic. *Annals of Pure and Applied Logic*, 162:419–446, 2011.

[20] J. Krajíček. Fragments of bounded arithmetic and bounded query classes. *Transactions of the American Mathematical Society*, 338:587–598, 1993.

[21] J. Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1995.

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

[23] I. Oitavem. *Três Assuntos de Lógica e Complexidade* (in Portuguese). Master thesis, Universidade de Lisboa, 1995.

[24] E. Silva. *Caracterização Aritmética em Primeira Ordem de Funções Computáveis em Espaço Polimonial* (in Portuguese). PhD thesis, Universidade Federal de Pernambuco, 2008.

[25] A. Skelley. A third-order bounded arithmetic theory for PSPACE. *CSL, Lecture Notes in Computer Science*, 3210:340–354, 2004.

[26] S. G. Simpson. *Subsystems of second order arithmetic*. Perspectives in Mathematical Logic. Springer, Berlin, 1999.

Received

Ricardo Bianconi, Departamento de Matemática, Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão, 1010, CEP 05508-090, São Paulo, SP, Brazil

E-mail: bianconi@ime.usp.br

Gilda Ferreira, Departamento de Matemática, Universidade Lusófona de Humanidades e Tecnologias, Av. do Campo Grande, 376, 1749-024 Lisboa, Portugal and CMAF-Universidade de Lisboa

E-mail: gildafer@cii.fc.ul.pt

Emmanuel Silva, Tribunal Regional Federal da 5ª Região, Cais do Apolo, s/n - Edifício Ministro Djaci Falcão, Bairro do Recife - Recife - PE, CEP 50030-908, Brazil

E-mail: eflsilva@trf3.jus.br