# Approximate counting in bounded arithmetic

Emil Jeřábek*
Institute of Mathematics, AS CR
jerabek@math.cas.cz

August 10, 2007

**Abstract**

We develop approximate counting of sets definable by Boolean circuits in bounded arithmetic using the dual weak pigeonhole principle ($dWPHP(PV)$), as a generalization of results from [15]. We discuss applications to formalization of randomized complexity classes (such as $BPP$, $APP$, $MA$, $AM$) in $PV_1 + dWPHP(PV)$.

## Introduction

One of the most important aspects of bounded arithmetic is its close connection to computational complexity. There is a correspondence between arithmetical theories, and complexity classes: Buss' theories $S^i_2$ and $T^i_2$ [6] correspond to levels of the polynomial-time hierarchy, and various second-order theories were constructed for weak classes such as $TC^0$; Cook [11] presents a uniform way of constructing "minimal theories" associated to complexity classes below $P$. Consequently, fundamental problems from complexity theory are tied to similar questions about the arithmetical theories; for instance, the hierarchy of Buss' theories collapses if and only if bounded arithmetic proves the collapse of the polynomial hierarchy.

Our main motivation for studying approximate counting is the problem whether we can associate theories to randomized complexity classes, like $BPP$ or $AM$. The problem is a loose research program rather than an exact question. On one hand, the concept of correspondence between theories and complexity classes does not admit a general definition; the way in which $T^1_2$ corresponds to $P^{NP}$ is rather different from the correspondence of $U^1$ to $NC$. On the other hand, many probabilistic classes like $BPP$ are "semantic classes", which means that attempts to characterize them as provably total functions of some kind in a recursively axiomatized theory are bound to failure. Nevertheless, we will try to provide evidence that $PV_1 + dWPHP(PV)$ (i.e., $PV_1$ extended by the dual (surjective) weak pigeonhole principle for poly-time computable functions) is the "right" theory for reasoning about randomized algorithms.

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

The connection of $dWPHP(PV)$ to probabilistic computation was first noticed by A. Wilkie, who proved that $\Sigma^b_1$-consequences of $S^1_2 + dWPHP(PV)$ are witnessed by *TFRP*-functions, and in particular, predicates provably $\Delta^b_1$ in $S^1_2 + dWPHP(PV)$ are in *ZPP* (the result was published in Krajíček [18]). Jeřábek [15] considered the converse problem of formalizing probabilistic algorithms in $S^1_2 + dWPHP(PV)$, and introduced a way to define *FRP*-functions in $S^1_2 + dWPHP(PV)$ which covered at least the witnessing functions from Wilkie's theorem; however, the method used was seemingly ad hoc, and it was not clear how it could be generalized to other complexity classes like *BPP*.

In this paper, we will show that the dual weak pigeonhole principle is strong enough to provide a general method of approximating probabilities. More precisely, if $X$ is a subset of an interval $[0, a)$ definable by a $PV$-formula, we can estimate $\Pr_{x<a}(x \in X)$ within a polynomially small error in $PV_1 + dWPHP(PV)$, and events of higher complexity can be dealt with by appropriate relativization. This allows us to treat various randomized classes like *BPP*, *APP*, *AM*, in a uniform and intuitive way—in fact, once we have a reasonable notion of (approximate) probability, the usual definitions of these classes can be formalized almost literally. As we have already mentioned, provably total functions are not an appropriate standard for establishing correspondence of theories to probabilistic complexity classes: for semantic classes there is no hope, and as we will see, for syntactic classes the problem is either meaningless or trivial (with the notable exception of *APP*). Instead, we will show that $PV_1 + dWPHP(PV)$ proves basic properties of the relevant probabilistic algorithms, such as amplification of success, or simulation of randomness by nonuniformity.

Estimating probabilities in uniform distributions is only a fancy name for approximate counting of bounded sets. Approximate counting has other applications besides randomized algorithms; most importantly, counting arguments are often used to prove various combinatorial theorems. We will provide basic counting tools like the inclusion-exclusion principle, but the overall utility of our methods in this area seems rather limited. Proofs of combinatorial statements such as the Ramsey theorem or the tournament principle typically rely on counting of sparse sets, which is impossible in our setup. We can only approximate the size of a set $X \subseteq [0, 2^n)$ within a polynomial fraction of $2^n$, whereas here we would need to approximate it within a polynomial fraction of $|X|$.

The paper is organized as follows. In section 1 we provide elementary background on basic arithmetic, and fix notational conventions. In section 2 we introduce approximate counting of sets defined by circuits in $PV_1 + dWPHP(PV)$, and formalize a toolbox of counting principles. In section 3 we discuss in detail the development of several randomized complexity classes (*FRP*, *BPP*, *APP*, *MA*, and promise variants) in $PV_1 + dWPHP(PV)$. In section 4 we indicate how to relativize our approach, and we discuss the class *AM*.

# 1 Preliminaries

We assume some degree of familiarity with first-order bounded arithmetic, however the basic definitions are summarized below. More background can be found in [18, 8, 13].

Buss' $S^i_2$ and $T^i_2$ [6] are first-order theories with equality in the language $L = \langle 0, S, +, \cdot, \leq,$

$\#, |x|, \lfloor \frac{x}{2} \rfloor \rangle$, where the function $|x|$ is intended to designate $\lceil \log_2(x+1) \rceil$ (the number of digits in the binary representation of $x$), and $x \# y$ is $2^{|x| \cdot |y|}$. *Bounded quantifiers* are expressions of the form

$$\exists x \le t \ldots := \exists x \, (x \le t \wedge \ldots),$$
$$\forall x \le t \ldots := \forall x \, (x \le t \to \ldots),$$

where $t$ is a term without an occurrence of $x$. A bounded quantifier is *sharply bounded*, if $t$ has the form $|s|$ for some term $s$. A formula $\varphi$ is sharply bounded, if all quantifiers in $\varphi$ are sharply bounded. The hierarchy of $\Sigma^b_i$- and $\Pi^b_i$-formulas is defined inductively: $\Sigma^b_0 = \Pi^b_0$ is the set of sharply bounded formulas, $\Sigma^b_{i+1}$ is the closure of $\Pi^b_i$ under bounded existential and sharply bounded universal quantifiers, and $\Pi^b_{i+1}$ is the closure of $\Sigma^b_i$ under bounded universal and sharply bounded existential quantifiers. Bounded formulas capture the polynomial-time hierarchy ($PH$). More precisely, for any $i \ge 1$ the class $\Sigma^P_i$ coincides with sets of natural numbers definable by $\Sigma^b_i$-formulas in $\mathbb{N}$ (the standard model of arithmetic), and dually $\Pi^P_i = \Pi^b_i(\mathbb{N})$, in particular $NP = \Sigma^b_1(\mathbb{N})$.

The theory $S^i_2$ consists of a finite list of open axioms denoted by $BASIC$, and the polynomial induction schema

$$(\Sigma^b_i\text{-}PIND) \qquad \varphi(0) \wedge \forall x \le a \, (\varphi(\lfloor \tfrac{x}{2} \rfloor) \to \varphi(x)) \to \varphi(a),$$

where $\varphi \in \Sigma^b_i$. The theory $T^i_2$ is axiomatized by $BASIC$ and the induction schema

$$(\Sigma^b_i\text{-}IND) \qquad \varphi(0) \wedge \forall x \le a \, (\varphi(x) \to \varphi(x+1)) \to \varphi(a).$$

$PV$ is a purely equational theory introduced by Cook [9]. Its language contains a few basic function symbols, and it is inductively expanded by symbols for functions defined from previously introduced functions by composition, and limited recursion on notation. $PV$ is axiomatized by equations defining all the function symbols, and a derivation rule similar to open *PIND*. In the standard model, $PV$-functions define exactly the class of polynomial-time computable functions ($FP$). We will slightly abuse the notation and denote by $PV$ also the language of $PV$ (the set of all $PV$-functions).

$PV_1$ (also called $QPV$) is an extension of $PV$ to first-order logic [19, 7, 10]. It has an axiomatization by purely universal sentences, and it is conservative over $PV$. The hierarchy of $\Sigma^b_i(PV)$- and $\Pi^b_i(PV)$-formulas is defined similarly to $\Sigma^b_i$ and $\Pi^b_i$, but in the language of $PV$. $PV_1$ proves *IND* and *PIND* for $\Sigma^b_0(PV)$-formulas.

$S^1_2(PV)$ is the combination of $S^1_2$ and $PV_1$: i.e., it has the language of $PV$, and it is axiomatized by $PV$ and $\Sigma^b_1(PV)$-*PIND*. All $PV$-functions have well-behaved provably total $\Delta^b_1$-definitions in $S^1_2$; it follows that $S^1_2(PV)$ is an extension of $S^1_2$ by definitions, and in particular, $S^1_2(PV)$ is conservative over $S^1_2$. Thus there is little practical difference between $S^1_2$ and $S^1_2(PV)$, and we will simply identify these two theories. Buss' witnessing theorem [6] implies that $S^1_2$ is $\Sigma^b_1$-conservative over $PV_1$, and in fact, we may identify $PV_1$ with $\forall\Sigma^b_1(S^1_2)$.

The theories $PV_{i+1}$ for $i > 0$, introduced in [19], are defined similarly to $PV_1$, except that the basic functions of their language include the characteristic functions of all $\Sigma^b_i$-predicates, thus $PV_{i+1}$-functions correspond to $FP^{\Sigma_i^P}$ in the standard model. $PV_{i+1}$ is a conservative extension of $T^i_2$ (contrary to popular belief, essentially the same also holds for $i = 0$ [16]), and $S^1_2(PV_{i+1})$ is a conservative extension of $S^{i+1}_2$. $S^{i+1}_2$ is $\Sigma^b_{i+1}$-conservative over $PV_{i+1}$ and $T^i_2$ by Buss' witnessing theorem.

All these theories can be *relativized*. We consider the language $L(\alpha) = L \cup \{\alpha\}$, where $\alpha$ is a new predicate, and define $\Sigma^b_i(\alpha)$ and $\Pi^b_i(\alpha)$ in the same way as $\Sigma^b_i$ and $\Pi^b_i$, but extended to the new language. The theories $S^i_2(\alpha)$ and $T^i_2(\alpha)$ are axiomatized by *BASIC* and $\Sigma^b_i(\alpha)$-*PIND* resp. $\Sigma^b_i(\alpha)$-*IND*, with no other axioms about $\alpha$. $PV(\alpha)$ and $PV_i(\alpha)$ can be defined similarly (the characteristic function of $\alpha$ is allowed to appear in functions constructed by limited recursion on notation). $PV(\alpha)$-functions correspond to polynomial-time algorithms with an oracle. We write $\varphi^\alpha$ and $f^\alpha$ when we want to stress the dependence of an $L(\alpha)$-formula or $PV(\alpha)$-function on $\alpha$; in that case, $\varphi^\psi$ or $f^\psi$ denotes the result of substitution of a formula $\psi$ for $\alpha$. We may generalize $L(\alpha)$ by allowing an arbitrary set of new predicates and function symbols instead of $\alpha$; in the case of functions, we have to include axioms enforcing an explicit polynomial bound on the length of the output of the function.

For any function $f$ we define the formula

$$dPHP^x_y(f) := \exists v < y\, \forall u < x\, f(u) \neq v,$$

where $f$ may involve other parameters not explicitly shown. The *dual* (or *surjective*) *weak pigeonhole principle* for $f$, written as $dWPHP(f)$, is the universal closure of the formula

$$x > 0 \to dPHP^{x|y|}_{x(|y|+1)}(f),$$

and if $\Gamma$ is a set of functions, $dWPHP(\Gamma)$ denotes the schema $\{dWPHP(f) \mid f \in \Gamma\}$. We will mostly work with $dWPHP(PV)$, i.e., the dual weak pigeonhole principle for poly-time functions. $dWPHP(PV)$ is over $S^1_2$ equivalent to the more usual schema

$$x > 1 \to dPHP^x_{x^2}(f),$$

but it is not clear whether this reduction also works over $PV_1$. $dWPHP(PV)$ is provable in $T^2_2$ [23, 18, 21], but $dWPHP(\alpha)$ is not provable in $S^2_2(\alpha)$ [24]. The schema $dWPHP(PV)$ is finitely axiomatizable: $PV_1$ proves that any $PV$-function is computable by a poly-size circuit on any bounded domain, thus $dWPHP(PV)$ is equivalent to its instance $dWPHP(\mathrm{eval})$, where $\mathrm{eval}(C, x)$ is a two-place $PV$-function which evaluates a circuit $C$ on an input $x$.

We will often work with *bounded definable sets*, which are collections of numbers of the form

$$X = \{x < a \mid \varphi(x)\},$$

where $\varphi$ is a formula. Bounded sets are *not* genuine objects in our arithmetical theories, but a figure of speech: $x \in X$ is an abbreviation for $x < a \wedge \varphi(x)$. When used in a context which asks for a set, a number $a$ is assumed to represent the integer interval $[0, a)$; thus, for example, $X \subseteq a$ means that all elements of $X$ are less than $a$. We will use simple set-theoretic operations, whose meaning should be generally clear from the context; for example, if $X \subseteq a$

and $Y \subseteq b$, we may define

$$X \times Y := \{bx + y \mid x \in X, y \in Y\} \subseteq ab,$$
$$X \dot{\cup} Y := X \cup \{y + a \mid y \in Y\} \subseteq a + b.$$

The sets we will encounter most often will be defined by *Boolean circuits*: a circuit $C\colon 2^n \to 2$ defines the set $\{x < 2^n \mid C(x) = 1\}$. (Here again, $2^n$ denotes the interval $[0, 2^n)$, which may be identified with the set of binary strings of length $n$; thus $C$ is a circuit with $n$ Boolean input variables.)

We will use the shorthand notation

$$x \in \mathrm{Log} \leftrightarrow \exists y\, x = |y|,$$
$$x \in \mathrm{LogLog} \leftrightarrow \exists y\, x = ||y||.$$

If $f$ is a function of two variables, $f(a, \bullet)$ denotes the function of one variable which results from $f$ by fixing its first argument to $a$. The set of natural numbers will be denoted by $\omega$ (in the metatheory).

We will also work with rational numbers in $PV$, which are assumed to be represented by pairs of integers in the natural way. The expression $x^{-1} \in \mathrm{Log}$ is a shorthand notation meaning that $x$ is a positive rational number, whose inverse is bounded from above by a natural number $n \in \mathrm{Log}$.

Many of our results take place *inside* formal theories like $PV_1 + dWPHP(PV)$. If $T$ is a theory, a parenthesized expression "in $T$" after the heading of a definition or theorem indicates that the definition is introduced in $T$, or that the theorem is formulated and proved inside $T$. However, we will slightly abuse this convention for reasons of compactness: when we write e.g. "for every $PV$-function $f$ ..." in a formalized context, it is assumed that the quantification over $PV$-functions takes place in the metatheory, and only *parameters* of the function are quantified inside $T$. Formulas, definable sets, and other non-first-order objects are treated similarly. Expressions like "a pair of $PV$-functions $\langle f, g\rangle$" also fit in this category; inside $T$, no actual pairing operation is involved.

## 2 Counting

Our definition of approximate counting in bounded arithmetic is based on the following observation: if $X$ and $Y$ are sets, and there exists a circuit which maps $X$ onto $Y$, then the cardinality of $Y$ is at most the cardinality of $X$. We need to make sure that such a definition is well-behaved, i.e., that it satisfies common properties we expect from a cardinality function. In particular, it is conceivable that a large but complicated set $X$ cannot be disentangled by a polynomial-size circuit and mapped onto an interval $[0, s)$ approaching its size; we must show that such cases do not happen. The natural way to guarantee sufficient precision of these counting circuits is to consider a two-sided comparison: if we find a mapping of $X$ onto $[0, s - e)$, and a mapping of $[0, s + e)$ onto $X$, we know that the size of $X$ is $s$ within error $e$.

It turns out that an extra complication is necessary: rather than mapping $X$ onto $Y$ directly, we will take several copies of both sets, i.e., map $v \times X$ onto $v \times Y$ for some $v > 0$.

With this modification, we are able to prove in $PV_1 + dWPHP(PV)$ that there exists a pair of counting circuits which estimates the size of $X$ within a polynomially small error (relative to the size of the ambient interval containing $X$), for any $X$ defined by a circuit. We will construct such counting circuits by analysis of the Nisan-Wigderson pseudorandom generator [22]; formalization of the Nisan-Wigderson generator in $S^1_2 + dWPHP(PV)$ was already considered in [15] for a different goal. We start by overview of the relevant concepts.

**Definition 2.1** (in $PV_1$) Let $f\colon 2^k \to 2$ be a truth-table of a Boolean function ($f$ is encoded as a string of $2^k$ bits, hence $k \in \mathrm{LogLog}$). We say that $f$ is (*worst-case*) *$\varepsilon$-hard,* written as $\mathrm{Hard}_\varepsilon(f)$, if there does not exist a circuit $C$ of size at most $2^{\varepsilon k}$ which computes $f$. The function $f$ is *average-case $\varepsilon$-hard,* written as $\mathrm{Hard}^A_\varepsilon(f)$, if there does not exist a circuit $C$ of size at most $2^{\varepsilon k}$ such that

$$\left|\{u < 2^k \mid C(u) = f(u)\}\right| \geq \left(\frac{1}{2} + 2^{-\varepsilon k}\right) 2^k.$$

Notice that $\mathrm{Hard}_\varepsilon(f)$ and $\mathrm{Hard}^A_\varepsilon(f)$ are $\Pi^b_1$-formulas.

**Lemma 2.2 ([15])** *For every constant $\varepsilon < 1/3$ there exists a constant $c$ such that $PV_1 + dWPHP(PV)$ proves: for every $k \in \mathrm{LogLog}$ such that $k \geq c$, there exist average-case $\varepsilon$-hard functions $f\colon 2^k \to 2$.*

*Moreover, there exists a $PV$-function $g\colon 2^{n-m} \to 2^n$ such that any $f < 2^n$ outside the range of $g$ is average-case $\varepsilon$-hard, where $n = 2^k$, and $m \geq n^{1-2\varepsilon}$.*

**Definition 2.3 ([22])** (in $PV_1$) Let $k, \ell, t, m \in \mathrm{Log}$, $k \leq \ell \leq t$. A *$\langle k, \ell, t, m\rangle$-design* is a sequence $\langle S_i\rangle_{i<m}$ of subsets $S_i \subseteq t$, such that $|S_i| = \ell$ and $|S_i \cap S_j| \leq k$ for all $i < j < m$.

**Lemma 2.4 ([15])** *Let $0 < \gamma < 1$. There are constants $\delta > 0$, $c > 1$, and a $PV$-function $d$ such that*

$$PV_1 \vdash d(x) \textit{ is a } \langle \gamma\ell, \ell, c\ell, 2^{\delta\ell}\rangle\textit{-design, where } \ell = ||x||.$$

**Definition 2.5 ([22])** (in $PV_1$) Let $x < 2^t$, and $X \subseteq t$, $|X| = \ell$. Let $\{s_i\}_{i<\ell}$ be the increasing enumeration of the set $X$. Then we put $x \restriction X := y$, where $y < 2^\ell$ and $\mathrm{bit}(y, i) = \mathrm{bit}(x, s_i)$ for all $i < \ell$.

If $f\colon 2^\ell \to 2$, and $S = \langle S_i\rangle_{i<m}$ is a $\langle k, \ell, t, m\rangle$-design, the *Nisan-Wigderson generator* is a function $NW_{f,S}\colon 2^t \to 2^m$ defined by

$$\mathrm{bit}(NW_{f,S}(x), i) = f(x \restriction S_i).$$

**Definition 2.6** (in $PV_1$) We adopt a few conventions on functions computed by circuits. Let $C\colon 2^n \to 2^m$ be a circuit, and $X$ and $Y$ definable sets. We say that $C$ *computes a function from $X$ to $Y$*, written as

$$C\colon X \to Y,$$

if $X \subseteq 2^n$, $Y \subseteq 2^m$, and $C[X] \subseteq Y$. We write

$$C\colon X \hookrightarrow Y$$

if, in addition, the function computed by $C$ is injective on $X$.

We write

$$C\colon X \twoheadrightarrow Y$$

if $X \subseteq 2^n$, $Y \subseteq 2^m$, and $C[X] \supseteq Y$. Notice that this does *not* imply $C\colon X \to Y$. An equivalent condition is $C\colon X' \to Y$ and $C[X'] = Y$ for some $X' \subseteq X$.

(This way of introducing $\twoheadrightarrow$ is mostly a technicality, needed to overcome the annoying fact that a non-empty set cannot be mapped onto the empty set.)

We are ready for the main theorem of this section, which guarantees the existence of suitable counting circuits. It is an extension of proposition 4.7 in [15].

**Theorem 2.7** *(in $PV_1 + dWPHP(PV)$) Let $C\colon 2^n \to 2$ be a Boolean circuit, and $\varepsilon^{-1} \in \mathrm{Log}$. Denote*

$$X := \{x < 2^n \mid C(x) = 1\}.$$

*There exist $s \le 2^n$, $v \le poly(n\varepsilon^{-1}|C|)$, and circuits $G_\xi, H_\xi$, $\xi = 0, 1$, of size $poly(n\varepsilon^{-1}|C|)$ such that*

$$G_0\colon v(s + \varepsilon 2^n) \twoheadrightarrow v \times X \qquad H_0\colon v \times X \hookrightarrow v(s + \varepsilon 2^n)$$
$$G_1\colon v \times (X \,\dot\cup\, \varepsilon 2^n) \twoheadrightarrow vs \qquad H_1\colon vs \hookrightarrow v \times (X \,\dot\cup\, \varepsilon 2^n)$$

*and such that*

$$G_\xi \circ H_\xi = \mathrm{id}$$

*on their respective domains.*

*Proof:* Let $\delta$ and $c$ be the constants from lemma 2.4 for $\gamma := 1/12$. Put

$$\ell := \max\bigl\{4|n\varepsilon^{-1}|, 12|n|, \frac{1}{\delta}|n|, 4(||C|| + 1)\bigr\},$$

and $k := \gamma\ell$, $t := c\ell$, $v := 2^t$. As $n \le 2^{\delta\ell}$, there exists a $\langle k, \ell, t, n\rangle$-design $S = \langle S_0, \dots, S_{n-1}\rangle$. By lemma 2.2, there exists an average-case 1/4-hard Boolean function $f\colon 2^\ell \to 2$. We define

$$Y := \{x < 2^t \mid C(NW_{f,S}(x)) = 1\},$$
$$s := 2^{n-t}|Y|.$$

(We may count $|Y|$ directly, as $t \in \mathrm{LogLog}$.)

For any $i \le n$, we define

$$M_i = \{\langle \vec{r}, x\rangle \in 2^n \times 2^t \mid C(f(x \restriction S_0), \dots, f(x \restriction S_{i-1}), r_i, \dots, r_{n-1}) = 1\}.$$

Notice that $M_0 = X \times 2^t$, and $M_n = 2^n \times Y$. Suppose we find a sequence of circuits $G_{\xi,i}$, $H_{\xi,i}$, where $\xi = 0, 1$ and $i < n$, such that

$$G_{0,i}\colon M_{i+1} \,\dot\cup\, (i+1)a2^{n+t-\ell} \twoheadrightarrow M_i \,\dot\cup\, ia2^{n+t-\ell}$$
$$H_{0,i}\colon M_i \,\dot\cup\, ia2^{n+t-\ell} \hookrightarrow M_{i+1} \,\dot\cup\, (i+1)a2^{n+t-\ell}$$
$$G_{1,i}\colon M_i \,\dot\cup\, (n-i)a2^{n+t-\ell} \twoheadrightarrow M_{i+1} \,\dot\cup\, (n-i-1)a2^{n+t-\ell}$$
$$H_{1,i}\colon M_{i+1} \,\dot\cup\, (n-i-1)a2^{n+t-\ell} \hookrightarrow M_i \,\dot\cup\, (n-i)a2^{n+t-\ell}$$
$$G_{\xi,i} \circ H_{\xi,i} = \mathrm{id}$$

where $a = 2^{3\ell/4}$. Then we can define

$$G_0 = G_{0,0} \circ G_{0,1} \circ \cdots \circ G_{0,n-1} \qquad H_0 = H_{0,n-1} \circ H_{0,n-2} \circ \cdots \circ H_{0,0}$$
$$G_1 = G_{1,n-1} \circ G_{1,n-2} \circ \cdots \circ G_{1,0} \qquad H_1 = H_{1,0} \circ H_{1,1} \circ \cdots \circ H_{1,n-1}$$

Notice that $v\varepsilon 2^n \geq na2^{n+t-\ell}$, as $n\varepsilon^{-1} \leq 2^{\ell/4}$. For any $x \in X \times 2^t$ and $y \in 2^n \times Y$, we can show

$$((G_{0,0} \circ G_{0,1} \circ \cdots \circ G_{0,i}) \circ (H_{0,i} \circ \cdots \circ H_{0,1} \circ H_{0,0}))(x) = x$$
$$((G_{1,n-1} \circ G_{1,n-2} \circ \cdots \circ G_{1,n-i}) \circ (H_{1,n-i} \circ \cdots \circ H_{1,n-2} \circ H_{1,n-1}))(y) = y$$

by straightforward induction on $i$, in particular $G_\xi \circ H_\xi = \mathrm{id}$, which also implies that $G_\xi$ are surjective, and $H_\xi$ are injective.

It thus suffices to construct $G_{\xi,i}$ and $H_{\xi,i}$. There exists an easily computable bijection between pairs $\langle y, u\rangle \in 2^{t-\ell} \times 2^\ell$, and numbers $x \in 2^t$, so that $x$ maps to $\langle x \restriction (t \smallsetminus S_i), x \restriction S_i\rangle$. If $j < n$, $y < 2^{t-\ell}$, $u < 2^\ell$, and $x < 2^t$ is such that $\langle y, u\rangle = \langle x \restriction (t \smallsetminus S_i), x \restriction S_i\rangle$, we define $f_j^{i,y}(u) = f(x \restriction S_j)$. Notice that $f_i^{i,y}(u) = f(u)$. Then

$$M_i \approx 2^i \times M_i',$$

where

$$M_i' := \{\langle r_{i+1}, \dots, r_{n-1}, y, r, u\rangle \in 2^{n-i-1} \times 2^{t-\ell} \times 2 \times 2^\ell \mid$$
$$C(f_0^{i,y}(u), \dots, f_{i-1}^{i,y}(u), r, r_{i+1}, \dots, r_{n-1}) = 1\},$$

and $A \approx B$ means that there exists a bijection $g$ of $A$ onto $B$ such that $g$ and $g^{-1}$ are computable by a polynomial-size circuit. In a similar way we have

$$M_{i+1} \approx 2^i \times M_{i+1}',$$

where

$$M_{i+1}' := \{\langle r_{i+1}, \dots, r_{n-1}, y, r, u\rangle \mid C(f_0^{i,y}(u), \dots, f_{i-1}^{i,y}(u), f(u), r_{i+1}, \dots) = 1\}.$$

Fix $y < 2^{t-\ell}$, and $r_{i+1}, \dots, r_{n-1} < 2$. Define

$$\begin{aligned}
U^{\vec{r},y} &:= \{\langle r, u\rangle \in 2 \times 2^\ell \mid C(f_0^{i,y}(u), \dots, f_{i-1}^{i,y}(u), r, r_{i+1}, \dots, r_{n-1}) = 1\} \\
&= \{\langle r, u\rangle \in 2 \times 2^\ell \mid \langle \vec{r}, y, r, u\rangle \in M_i'\}, \\
V^{\vec{r},y} &:= \{\langle r, u\rangle \in 2 \times 2^\ell \mid C(f_0^{i,y}(u), \dots, f_{i-1}^{i,y}(u), f(u), r_{i+1}, \dots, r_{n-1}) = 1\} \\
&= \{\langle r, u\rangle \in 2 \times 2^\ell \mid \langle \vec{r}, y, r, u\rangle \in M_{i+1}'\}, \\
A_\eta(u) &:= C(f_0^{i,y}(u), \dots, f_{i-1}^{i,y}(u), \eta, r_{i+1}, \dots, r_{n-1}),
\end{aligned}$$

where $\eta < 2$. As $\ell \in \mathrm{LogLog}$, we can directly count the sets $U^{\vec{r},y}$ and $V^{\vec{r},y}$; an easy calculation shows

$$\begin{aligned}
|V^{\vec{r},y}| - |U^{\vec{r},y}| &= 2\big|\{u \mid f(u) \wedge A_1(u)\}\big| + 2\big|\{u \mid \neg f(u) \wedge A_0(u)\}\big| \\
&\quad - \big|\{u \mid A_1(u)\}\big| - \big|\{u \mid A_0(u)\}\big| \\
&= \big|\{u \mid f(u) \wedge A_1(u)\}\big| - \big|\{u \mid \neg f(u) \wedge A_1(u)\}\big| \\
&\quad + \big|\{u \mid \neg f(u) \wedge A_0(u)\}\big| - \big|\{u \mid f(u) \wedge A_0(u)\}\big| \\
&= \big|\{u \mid f(u) \wedge A_1(u)\}\big| + \big|\{u \mid \neg f(u) \wedge \neg A_1(u)\}\big| \\
&\quad + \big|\{u \mid \neg f(u) \wedge A_0(u)\}\big| + \big|\{u \mid f(u) \wedge \neg A_0(u)\}\big| - 2^{\ell} \\
&= \big|\{u \mid f(u) \leftrightarrow A_1(u)\}\big| - \big|\{u \mid f(u) \leftrightarrow A_0(u)\}\big|
\end{aligned}$$

On the other hand, for any $j \neq i$, $f_j^{i,y}(u)$ depends only on $|S_i \cap S_j| \leq k$ variables of $u$, and is thus computable by a circuit of size $2^k$. Therefore, $A_\eta$ and $\neg A_\eta$ are computable by circuits of size at most

$$1 + |C| + i2^k \leq |C| + n2^k \leq 2^{\ell/4-1} + 2^{\ell/12}2^{\ell/12} \leq 2^{\ell/4}.$$

As $f$ is average-case 1/4-hard, we have

$$\big||\{u \mid A_\eta(u) = f(u)\}| - 2^{\ell-1}\big| \leq 2^{\ell-\ell/4} = a,$$

thus

$$\big||V^{\vec{r},y}| - |U^{\vec{r},y}|\big| \leq 2a.$$

We may arrange the sets $U^{\vec{r},y}$ and $V^{\vec{r},y}$ in increasing sequences, match their initial parts, and pad to get functions

$$\begin{aligned}
&g_0^{\vec{r},y} \colon U^{\vec{r},y} \mathbin{\dot\cup} 2a \twoheadrightarrow V^{\vec{r},y} &\qquad& h_0^{\vec{r},y} \colon V^{\vec{r},y} \hookrightarrow U^{\vec{r},y} \mathbin{\dot\cup} 2a \\
&g_1^{\vec{r},y} \colon V^{\vec{r},y} \mathbin{\dot\cup} 2a \twoheadrightarrow U^{\vec{r},y} && h_1^{\vec{r},y} \colon U^{\vec{r},y} \hookrightarrow V^{\vec{r},y} \mathbin{\dot\cup} 2a
\end{aligned}$$

such that $g_\xi^{\vec{r},y} \circ h_\xi^{\vec{r},y} = \mathrm{id}$. As this construction is uniform in $\vec{r}$ and $y$, we may construct polynomial-size circuits

$$\begin{aligned}
&G_0' \colon M_i' \mathbin{\dot\cup} a2^{n-i+t-\ell} \twoheadrightarrow M_{i+1}' &\qquad& H_0' \colon M_{i+1}' \hookrightarrow M_i' \mathbin{\dot\cup} a2^{n-i+t-\ell} \\
&G_1' \colon M_{i+1}' \mathbin{\dot\cup} a2^{n-i+t-\ell} \twoheadrightarrow M_i' && H_1' \colon M_i' \hookrightarrow M_{i+1}' \mathbin{\dot\cup} a2^{n-i+t-\ell}
\end{aligned}$$

and from these we obtain $G_{\xi,i}$, $H_{\xi,i}$ as required. $\square$

We formally introduce the concept of approximate size comparison, as described in the introductory paragraph of this section. Notice that the definition applies to a more general situation than what is permitted by theorem 2.7. The main reason is that we will occasionally need to express that a set is exponentially small, even though theorem 2.7 cannot provide counting with exponential precision.

**Definition 2.8** (in $PV_1 + dWPHP(PV)$) Let $X, Y \subseteq 2^n$ be definable sets, and $\varepsilon \leq 1$. We say that *the size of $X$ is approximately less than the size of $Y$ with error $\varepsilon$*, written as

$$X \preceq_\varepsilon Y,$$

if there exists a circuit $G$, and $v \neq 0$, such that

$$G\colon v \times (Y \mathbin{\dot\cup} \varepsilon 2^n) \twoheadrightarrow v \times X.$$

The sets $X$ and $Y$ have *approximately the same size with error* $\varepsilon$, written as

$$X \approx_\varepsilon Y,$$

if $X \preceq_\varepsilon Y$ and $Y \preceq_\varepsilon X$.

We recall that we identify a number $s$ with the interval $[0, s)$, thus as a special case, $X \approx_\varepsilon s$ means that the size of $X$ is equal to $s$ with error $\varepsilon$.

**Remark 2.9** In this definition, "error $\varepsilon$" is somewhat a misnomer. The counting is not exact even if we take $\varepsilon = 0$, there is always some error present due to the fact that only the weak pigeonhole principle is available. In fact, we will often conveniently use $\preceq_0$ for approximate size comparisons.

The lemma below summarizes elementary properties of definition 2.8.

**Lemma 2.10** *(in $PV_1$) Let $X, Y, X', Y', Z \subseteq 2^n$ and $W, W' \subseteq 2^m$ be definable sets, and $\varepsilon, \delta \leq 1$.*

*(i)* $X \preceq_\varepsilon Y$, $\varepsilon \leq \delta \Rightarrow X \preceq_\delta Y$.

*(ii)* $X \subseteq Y \Rightarrow X \preceq_0 Y$.

*(iii)* $X \preceq_\varepsilon Y$, $Y \preceq_\delta Z \Rightarrow X \preceq_{\varepsilon+\delta} Z$.

*(iv) If* $X \preceq_\varepsilon X'$, $Y \preceq_\delta Y'$, *and* $X'$ *and* $Y'$ *are separable by a circuit, then* $X \cup Y \preceq_{\varepsilon+\delta} X' \cup Y'$.

*(v)* $X \preceq_\varepsilon X'$, $W \preceq_\delta W' \Rightarrow X \times W \preceq_{\varepsilon+\delta+\varepsilon\delta} X' \times W'$.

*Proof:* Exercise. □

The next lemma exploits consequences of theorem 2.7.

**Lemma 2.11** *(in $PV_1 + dWPHP(PV)$) Let $X, Y \subseteq 2^n$ be definable by circuits, $s, t, u \leq 2^n$, $\varepsilon, \delta, \eta, \xi \leq 1$, $\xi^{-1} \in \mathrm{Log}$.*

*(i) There exists* $s \leq 2^n$ *such that* $X \approx_\xi s$.

*(ii)* $s \preceq_\varepsilon X \preceq_\delta t \Rightarrow s \leq t + (\varepsilon + \delta + \xi)2^n$.

*(iii)* $X \preceq_\xi Y$ *or* $Y \preceq_\xi X$.

*(iv)* $X \preceq_\varepsilon Y \Rightarrow 2^n \smallsetminus Y \preceq_{\varepsilon+\xi} 2^n \smallsetminus X$.

*(v)* $X \approx_\varepsilon s$, $Y \approx_\delta t$, $X \cap Y \approx_\eta u \Rightarrow X \cup Y \approx_{\varepsilon+\delta+\eta+\xi} s + t - u$.

*Proof:* $(i)$ follows from theorem 2.7.

$(ii)$: by transitivity, it suffices to show that $s \preceq_0 t$ implies $s \le t + \xi 2^n$, which follows from $dWPHP(PV)$.

$(iii)$ follows from $(i)$, and linearity of $\le$.

$(iv)$: let $\zeta = \xi/11$, and choose $s, t, s', t'$ such that $X \approx_\zeta s$, $Y \approx_\zeta t$, $2^n \smallsetminus X \approx_\zeta s'$, $2^n \smallsetminus Y \approx_\zeta t'$. We have $s \le t + (\varepsilon + 3\zeta)2^n$ by $(ii)$. As $t + t' \preceq_{2\zeta} 2^n$ by lemma 2.10 $(iv)$, we have also $t' \le 2^n - t + 3\zeta 2^n$ by $(ii)$, and in a similar way, $2^n - s \le s' + 3\zeta 2^n$. This implies $t' \le s' + (\varepsilon + 9\zeta)2^n$, thus $2^n \smallsetminus Y \preceq_{\varepsilon+11\zeta} 2^n \smallsetminus X$.

$(v)$: fix $r$ such that $X \smallsetminus Y \approx_{\xi/2} r$. By lemma 2.10 $(iv)$, we have $X \approx_{\eta+\xi/2} r + u$, and $X \cup Y \approx_{\delta+\xi/2} r + t$. The former implies $s \approx_{\varepsilon+\eta+\xi/2} r + u$, thus $s + t - u \approx_{\varepsilon+\eta+\xi/2} r + t$, and $s + t - u \approx_{\varepsilon+\delta+\eta+\xi} X \cup Y$. □

The definition of $\preceq_\varepsilon$ is problematic, if we wish to use it in induction formulas in more sophisticated arguments. As it stands, it is an unbounded $\exists\Pi^b_2$-formula; even if we restrict its usage to the case covered by theorem 2.7, and include the relevant bounds, we cannot do much better than $\Sigma^b_2$. We can solve this problem by working in a suitable conservative extension of $PV_1 + dWPHP(PV)$, introduced in [15].

**Definition 2.12** The theory $HARD^A$ is an extension of $PV_1(\alpha) + dWPHP(PV(\alpha))$ by the axioms

$$\alpha(x) \text{ is a truth-table of a Boolean function in } ||x|| \text{ variables,}$$
$$x \ge c \to \mathrm{Hard}^A_{1/4}(\alpha(x)),$$
$$||x|| = ||y|| \to \alpha(x) = \alpha(y),$$

where $c$ is the constant from lemma 2.2.

**Theorem 2.13** *$HARD^A$ is a conservative extension of $PV_1 + dWPHP(PV)$. More generally, for any $i \ge 1$, $HARD^A + S^i_2(\alpha)$ and $HARD^A + T^i_2(\alpha)$ are conservative extensions of $S^i_2 + dWPHP(PV)$ and $T^i_2 + dWPHP(PV)$, respectively.*

*Proof:* This was shown in [15] with $S^1_2$ as a base theory. It is easy to modify the proof so that it works over $PV_1$. □

We note that the axiom $dWPHP(PV(\alpha))$ is redundant in $HARD^A + S^1_2(\alpha)$; i.e., the existence of functions hard on average implies $dWPHP(PV)$ over $S^1_2$ [15]. We do not know whether this also holds over $PV_1$.

**Lemma 2.14** *There is a $PV(\alpha)$-function* Size *such that $HARD^A$ proves: if $X \subseteq 2^n$ is definable by a circuit $C$, then*

$$X \approx_\varepsilon \mathrm{Size}(C, 2^n, e),$$

*where $\varepsilon = |e|^{-1}$. The "witnessing circuits" $G_\xi, H_\xi$ from theorem 2.7 are also constructible by $PV(\alpha)$-functions.*

*Proof:* By inspection of the proof of theorem 2.7, we see that the only non-uniformity was in the choice of the hard function $f$. □

We will abuse the notation and write $\mathrm{Size}(X,\varepsilon)$ instead of $\mathrm{Size}(C,2^n,e)$.

The advantage of $HARD^A$ is that the complexity of approximate counting drops from $\Sigma^b_2$ to $PV(\alpha)$, which means that we can use approximate counting freely in induction, and we can count parametric families of sets uniformly. Some of the results below illustrate these techniques. We begin by showing that the size of the disjoint union of a sequence of sets is the sum of sizes of the sets.

**Proposition 2.15 (Disjoint union)** *(in $PV_1 + dWPHP(PV)$) Let $\{X_i \mid i < m\}$ be subsets of $2^n$, defined by a sequence of circuits. Let $\varepsilon, \xi \le 1$, $\xi^{-1} \in \mathrm{Log}$, and $\{s_i \mid i < m\}$ a sequence of numbers such that $X_i \preceq_\varepsilon s_i$ for every $i < m$. Then*

$$\sum_{i<m} X_i \preceq_{\varepsilon+\xi} \sum_{i<m} s_i,$$

*where the disjoint sum $\sum_{i<m} X_i := \bigcup_{i<m}(X_i \times \{i\}) \subseteq 2^n \times m$ is considered a subset of $2^{n+|m|}$.*

*The same holds for $\succeq$ in place of $\preceq$.*

*Proof:* We may work in $HARD^A$ by theorem 2.13. First, notice that the error in $\preceq$ is relative to the ambient set size, thus if we reconsider $X_i$ as a subset of $2^n \times m$, we have $X_i \preceq_{\varepsilon/m} s_i$. Put $\zeta = \xi/(3m+1)$. We will show

$$\mathrm{Size}\Big(\sum_{i<k} X_i, \zeta\Big) \le \sum_{i<k} s_i + (\varepsilon/m + 3\zeta)k$$

by induction on $k \le m$. Assume that the statement is true for $k$. We have

$$\sum_{i<k} X_i \approx_\zeta \mathrm{Size}\Big(\sum_{i<k} X_i, \zeta\Big) \preceq_\delta \sum_{i<k} s_i,$$

where $\delta = (\varepsilon/m + 3\zeta)k$. As $X_k \preceq_{\varepsilon/m} s_k$, we obtain

$$\mathrm{Size}\Big(\sum_{i\le k} X_i, \zeta\Big) \approx_\zeta \sum_{i\le k} X_i \preceq_{\varepsilon/m+\zeta+\delta} \sum_{i\le k} s_i$$

by lemma 2.10 $(iv)$, thus

$$\mathrm{Size}\Big(\sum_{i\le k} X_i, \zeta\Big) \le \sum_{i\le k} s_i + (\varepsilon/m + 3\zeta)(k+1)$$

by lemma 2.11 $(ii)$.

For $k = m$, we get

$$\sum_{i<m} X_i \approx_\zeta \mathrm{Size}\Big(\sum_{i<m} X_i, \zeta\Big) \preceq_{\varepsilon+3m\zeta} \sum_{i<m} s_i. \qquad \square$$

We can apply proposition 2.15 only to sequences of sets encoded by a number, in particular, the length of the sequence is in Log. We present a variant which applies to larger families of sets, whose sizes are uniformly bounded. We can also read it contrapositively as an averaging argument: if we have a family of at most $t$ sets, such that the size of their union is more than $st$, then one of the sets must be larger than $st/t = s$.

**Proposition 2.16 (Averaging)** *(in $PV_1 + dWPHP(PV)$) Let $X \subseteq 2^n \times 2^m$ and $Y \subseteq 2^m$ be definable by circuits, $Y \preceq_\delta t$, and $X_y \preceq_\varepsilon s$ for every $y \in Y$, where $X_y := \{x \mid \langle x, y\rangle \in X\}$. Then*

$$X \cap (2^n \times Y) \preceq_{\varepsilon+\delta+\varepsilon\delta+\xi} st$$

*for any $\xi^{-1} \in \mathrm{Log}$.*

*Proof:* By lemma 2.14, there are $PV(\alpha)$-functions $f, v$ such that

$$f(y, \bullet)\colon v(y) \times (\mathrm{Size}(X_y, \xi) + \xi 2^n) \twoheadrightarrow v(y) \times X_y.$$

We may easily arrange $v(y) = v$ to be independent on $y$, while increasing the error slightly. Also, if $y \in Y$, we have $\mathrm{Size}(X_y, \xi) \leq s + (\varepsilon + \xi)2^n$, thus we obtain a function $f'$ such that

$$f'(y, \bullet)\colon v \times (s + (\varepsilon + 3\xi)2^n) \twoheadrightarrow v \times X_y$$

for every $y \in Y$. There is a function $g$ and number $w$ such that

$$g\colon w \times (t + \delta 2^m) \twoheadrightarrow w \times Y,$$

and suitable composition of $g$ with $f'$ gives a function

$$vw(t + \delta 2^m)(s + 3\xi 2^n) \twoheadrightarrow vw \times (X \cap (2^n \times Y)).$$

We have

$$(t + \delta 2^m)(s + 3\xi 2^n) \leq st + (\varepsilon + \delta + \varepsilon\delta + 6\xi)2^{n+m},$$

thus $X \cap (2^n \times Y) \preceq_{\varepsilon+\delta+\varepsilon\delta+6\xi} st$. □

The next task is to formalize a suitable version of Chernoff's bound, which is *sine qua non* for development of randomized algorithms. The proof consists of two parts. The number-theoretic part is a bound on certain sums of binomial coefficients; we reduce it to a special case which was formalized in [15]. The combinatorial part of Chernoff's bound relies on the fact that we can construct counting circuits for a set $X$ and its complement $2^n \smallsetminus X$ so that the sizes approximately add up to $2^n$.

**Lemma 2.17** *There is a constant $c$ such that $PV_1$ proves: for any $n > 0$, $x > 0$, $y \leq x$, and $\delta, \varepsilon \in [0, 1]$, such that $n \in \mathrm{Log}$,*

$$\sum_{j \leq n(\frac{y}{x} - \delta)} \binom{n}{j} (y + \varepsilon x)^j (x - y + \varepsilon x)^{n-j} \leq c\, x^n 4^{n(c\varepsilon - \delta^2)}.$$

*Proof:* Put

$$k := \left\lfloor n\frac{y+\varepsilon x}{(1+2\varepsilon)x}\right\rfloor, \qquad i := k - \left\lfloor n\left(\frac{y}{x}-\delta\right)\right\rfloor.$$

We assume $k > i \geq 0$, the remaining borderline cases are left as an exercise. The left-hand side is at most

$$S := \sum_{j\leq k-i}\binom{n}{j}(k+1)^j(n-k)^{n-j}\left(\frac{x(1+2\varepsilon)}{n}\right)^n \leq c\,x^n\left(1+\frac{1}{k}\right)^k(1+2\varepsilon)^n 4^{-i^2/n}$$

by proposition A.5 in [15]. We also have

$$\left(1+\frac{1}{k}\right)^k \leq 4.$$

Assume for simplicity $\varepsilon \leq 1/4$, and put $\ell := \lfloor 1/(2\varepsilon)\rfloor$. Then

$$(1+2\varepsilon)^n \leq \left(1+\frac{1}{\ell}\right)^n \leq \left(1+\frac{1}{\ell}\right)^{\ell\lceil n/\ell\rceil} \leq 4^{\lceil n/\ell\rceil},$$

and

$$\frac{n}{\ell} \leq \frac{n}{1/(2\varepsilon)-1} = \frac{2\varepsilon n}{1-2\varepsilon} \leq 4\varepsilon n,$$

thus $(1+2\varepsilon)^n \leq 4\cdot 4^{4\varepsilon n}$.

We have

$$n\frac{y}{x} - k \leq 1 + n\left(\frac{y}{x} - \frac{y+\varepsilon x}{(1+2\varepsilon)x}\right) = 1 + n\frac{\varepsilon(2y-x)}{(1+2\varepsilon)x} \leq 1 + n\varepsilon,$$

thus

$$i \geq k - n\left(\frac{y}{x}-\delta\right) = \delta n - \left(n\frac{y}{x} - k\right) \geq \delta n - (1+\varepsilon n),$$

and

$$-\frac{i^2}{n} \leq -\frac{(\delta n - (1+\varepsilon n))^2}{n} \leq -\delta^2 n + 2\delta(1+\varepsilon n) \leq 2 - \delta^2 n + 2\varepsilon n.$$

Putting everything together, we have

$$S \leq 4^4 c\,x^n 4^{6\varepsilon n - \delta^2 n}. \qquad \square$$

**Proposition 2.18 (Chernoff's bound)** *(in $PV_1 + dWPHP(PV)$) Let $X \subseteq 2^n$ be defined by a circuit, $m \in \mathrm{Log}$, $0 \leq \varepsilon, \delta, p \leq 1$, and $X \succeq_\delta p2^n$. Then*

$$\left\{w \in (2^n)^m \;\middle|\; |\{i < m \mid w_i \in X\}| \leq m(p-\varepsilon)\right\} \preceq_0 c4^{m(c\delta-\varepsilon^2)}2^{nm}$$

*for some constant $c$, where $w$ is treated as a sequence of $m$ numbers less that $2^n$, and $w_i$ is its $i$th member.*

*Proof:* Let $\xi = 1/m$, and $s = \mathrm{Size}(X, \xi)$. There is a $v > 0$ and functions $f, g$ such that

$$f\colon v(s + \xi 2^n) \twoheadrightarrow v \times X,$$
$$g\colon v(2^n - s + \xi 2^n) \twoheadrightarrow v \times (2^n \smallsetminus X).$$

We can construct a function $h$ by taking $f$ and $g$ coordinatewise so that

$$h(I, \bullet)\colon v^m (s + \xi 2^n)^j (2^n - s + \xi 2^n)^{m-j} \twoheadrightarrow v^m \times \{w \mid I = \{i < m \mid w_i \in X\}\}$$

for every $I \subseteq m$ of size $j$. (The straightforward way of showing the surjectivity of $h$ uses $BB\Sigma^b_1$ to collect preimages under $f$ or $g$ into a sequence. The choice schema $BB\Sigma^b_1$ is not available in $PV_1$, but we can avoid it as $f$ and $g$ have coretractions computable by poly-size circuits by theorem 2.7.) We combine $h$ with enumeration of small subsets of $m$, and obtain a function

$$v^m \sum_{j \le m(p-\varepsilon)} \binom{m}{j} (s + \xi 2^n)^j (2^n - s + \xi 2^n)^{m-j} \twoheadrightarrow$$
$$\twoheadrightarrow v^m \times \bigl\{ w \in (2^n)^m \bigm| |\{i < m \mid w_i \in X\}| \le m(p - \varepsilon) \bigr\}.$$

Notice that $p2^n \le s + (\delta + 2\xi)2^n$ by lemma 2.11 (*ii*). We invoke lemma 2.17 with "$x$" $= 2^n$, "$y$" $= s + (\delta + 2\xi)2^n$, and "$\delta$" $= 3\xi + \delta$, which gives

$$\sum_{j \le m(p-\varepsilon)} \binom{m}{j} (s + \xi 2^n)^j (2^n - s + \xi 2^n)^{m-j} \le c 2^{nm} 4^{m(3c/m + c\delta - \varepsilon^2)} = 64^c c 2^{nm} 4^{m(c\delta - \varepsilon^2)}.$$

□

Another widely used property of counting is the inclusion-exclusion principle, which we formalize below. Notice that the assumptions on $k$ and $m$ are necessary so that the bounded sum in the statement of the principle is well-defined; thus it is not an additional restriction on applicability of the principle.

**Proposition 2.19 (Inclusion-exclusion principle)** (*in* $PV_1 + dWPHP(PV)$) *Let* $\{X_i \mid i < m\}$ *be subsets of* $2^n$*, defined by a sequence of circuits. Let* $k \le m$ *be such that* $k \in \mathrm{LogLog}$ *and* $(m/k)^k \in \mathrm{Log}$*. Assume*

$$\bigcap_{i \in I} X_i \approx_{\varepsilon_I} s_I$$

*for every* $I \subseteq m$ *of size at most* $k$*, and define*

$$s = \sum_{\substack{I \subseteq m \\ 0 < |I| \le k}} (-1)^{|I|+1} s_I, \qquad \varepsilon = \sum_{\substack{I \subseteq m \\ 0 < |I| \le k}} \varepsilon_I.$$

*Then*

$$\bigcup_{i < m} X_i \succeq_{\varepsilon + \xi} s$$

*if $k$ is even, and*

$$\bigcup_{i<m} X_i \preceq_{\varepsilon+\xi} s$$

*if $k$ is odd, for any $\xi^{-1} \in \mathrm{Log}$.*

*Proof:* The sums are well-defined, as

$$\binom{m}{\le k} := \sum_{i\le k} \binom{m}{i} \le (4m/k)^k \in \mathrm{Log}$$

can be shown by easy induction on $k$, using $(1+1/k)^k \le 4$. For any $i \le \ell < m$, we define

$$X_i^\ell := \begin{cases} X_i, & i<\ell, \\ \bigcup_{j=\ell}^{m-1} X_j, & i=\ell. \end{cases}$$

Assume $k>0$ is even, the case of odd $k$ is similar. Let $\eta^{-1} \in \mathrm{Log}$. We will show

$$\mathrm{Size}\Big(\bigcup_{i<m} X_i, \eta\Big) + 5\eta\binom{\ell}{\le k}2^n \ge \sum_{\substack{I\subseteq \ell+1 \\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i^\ell, \eta\Big)$$

by induction on $\ell < m$. The base case $\ell = 0$ is trivial. Assume that the statement holds for $\ell - 1$. We have

$$(*) := \sum_{\substack{I\subseteq \ell \\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i^{\ell-1}, \eta\Big) =$$

$$\sum_{\substack{I\subseteq \ell-1 \\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i, \eta\Big) - \sum_{\substack{I\subseteq \ell-1 \\ 0\le|I|<k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i \cap X_{\ell-1}^{\ell-1}, \eta\Big).$$

By lemma 2.11 $(v)$, we have

$$\begin{aligned}\mathrm{Size}\Big(\bigcap_{i\in I} X_i \cap X_{\ell-1}^{\ell-1}, \eta\Big) &= \mathrm{Size}\Big(\big(\bigcap_{i\in I\cup\{\ell-1\}} X_i^\ell\big) \cup \big(\bigcap_{i\in I\cup\{\ell\}} X_i^\ell\big), \eta\Big) \\ &= \mathrm{Size}\Big(\bigcap_{i\in I\cup\{\ell-1\}} X_i^\ell, \eta\Big) + \mathrm{Size}\Big(\bigcap_{i\in I\cup\{\ell\}} X_i^\ell, \eta\Big) \\ &\quad - \mathrm{Size}\Big(\bigcap_{i\in I\cup\{\ell-1,\ell\}} X_i^\ell, \eta\Big) \pm 5\eta 2^n,\end{aligned}$$

thus

$$\begin{aligned}(*) + 5\eta\binom{\ell-1}{\le k-1}2^n &\ge \sum_{\substack{I\subseteq \ell+1 \\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i^\ell, \eta\Big) + (-1)^k \sum_{\substack{I\subseteq \ell-1 \\ |I|=k-1}} \mathrm{Size}\Big(\bigcap_{i\in I\cup\{\ell-1,\ell\}} X_i^\ell, \eta\Big) \\ &\ge \sum_{\substack{I\subseteq \ell+1 \\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i^\ell, \eta\Big).\end{aligned}$$

Using the induction hypothesis, we get

$$\mathrm{Size}\Big(\bigcup_{i<m} X_i,\eta\Big) + 5\eta\left(\binom{\ell-1}{\le k} + \binom{\ell-1}{\le k-1}\right)2^n \ge \sum_{\substack{I\subseteq \ell+1\\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i^\ell,\eta\Big),$$

and we can easily derive

$$\binom{\ell-1}{\le k} + \binom{\ell-1}{\le k-1} = \binom{\ell}{\le k}$$

from $\binom{\ell}{i+1} = \binom{\ell-1}{i+1} + \binom{\ell-1}{i}$.

We take $\ell = m-1$. We have

$$\sum_{\substack{I\subseteq m\\ 0<|I|\le k}} (-1)^{|I|+1}\,\mathrm{Size}\Big(\bigcap_{i\in I} X_i,\eta\Big) + \left(\varepsilon + 2\eta\binom{m}{\le k}\right)2^n \ge s$$

by lemma 2.11 (*ii*), thus

$$\bigcup_{i<m} X_i \succeq_{\varepsilon+\xi} s,$$

where $\xi \le 7\eta\binom{m}{\le k}$. As $\binom{m}{\le k} \in \mathrm{Log}$, we can make $\xi$ arbitrarily small by choosing a suitable $\eta^{-1} \in \mathrm{Log}$. □

Approximate counting, and estimation of probability with respect to the uniform distribution are two sides of the same coin, thus we can introduce probabilities in $PV_1 + dWPHP(PV)$ as in the following definition. All the results of section 2 can be naturally restated in probabilistic terms, which we leave to reader's imagination.

**Definition 2.20** (in $PV_1 + dWPHP(PV)$) Let $X$ be a definable subset of $2^{|t|}$, and $0 \le \varepsilon, p \le 1$. We define

$$\Pr_{x<t}(x \in X) \preceq_\varepsilon p \quad \text{iff} \quad X \cap t \preceq_\varepsilon pt,$$

and similarly for $\succeq$, $\approx$. If $X$ is defined by a circuit and $\varepsilon^{-1} \in \mathrm{Log}$, we put

$$\Pr_{x<t}(x \in X)_\varepsilon := \frac{1}{t}\,\mathrm{Size}(X \cap t, \varepsilon).$$

# 3 Randomized algorithms

Our main application of approximate counting is in the formalization of probabilistic algorithms in $PV_1 + dWPHP(PV)$. We will consider in turn the classes *FRP*, *BPP*, *APP*, *MA*, including their promise versions (*prBPP*, *prMA*). For each class we present a natural way to define algorithms from the class in $PV_1 + dWPHP(PV)$ (and its extensions), and we prove in $PV_1 + dWPHP(PV)$ basic properties of the class (such as success amplification, or simulation by circuits). We also discuss the problem whether all algorithms from the class can be defined in $PV_1 + dWPHP(PV)$: in general, algorithms from "syntactic classes" (like *prBPP* or *APP*) are always definable, whereas "semantic classes" (like *BPP*) cannot be shown to be captured by $PV_1 + dWPHP(PV)$ (or in fact, any recursively axiomatizable theory), without nontrivial progress in their derandomization. In the case of semantic classes we pinpoint the problem by showing that definability of any particular algorithm is equivalent to provability of a $\forall\Sigma^b_1$-sentence. (We show that the class $APP$ is recursively enumerable, thus it can be considered a syntactic class even if that is not apparent from its definition.)

## 3.1 $NP$ search problems

The first class of algorithms we mention are probabilistic solvers to $NP$ search problems.

**Definition 3.1** An *$NP$ search problem $S$* is given by a poly-time computable relation $R(x, y)$ such that

$$R(x, y) \Rightarrow |y| \leq p(|x|)$$

for some polynomial $p$. Any $y$ such that $R(x, y)$ is a *solution* of $S$ for $x$, and $x$ is called an *instance* of $S$ in this context. The search problem $S$ is *total* if every instance of $S$ has a solution.

A deterministic algorithm *solves $S$* if it computes a solution for any given solvable instance of $S$. A probabilistic algorithm $A$ *solves $S$* if

$$\Pr(A(x) \text{ is a solution for } x) \geq 1/2$$

for every solvable instance $x$. (The constant $1/2$ is rather arbitrary.)

The class of $NP$ search problems solvable in probabilistic polynomial time is called $FRP$. The class of total search problems from $FRP$ is denoted $TFRP$.

Notice that we may require without loss of generality that an algorithm solving an $NP$ search problem rejects all unsolvable instances. The class of randomized poly-time algorithms which solve $NP$ search problems under this requirement can be defined directly, without any reference to search problems: a probabilistic algorithm $A$ computes an $FRP$-function, if for every input $x$, either $A(x)$ rejects with probability 1, or accepts and outputs a value with probability at least $1/2$. $FRP$ can thus be thought of as a class of partial multifunctions. Notice that a language $L$ is in $ZPP$ iff its characteristic function is in $FRP$, and $L \in RP$ iff it is the domain of an $FRP$-function, thus $FRP$ generalizes the classes $ZPP$ and $RP$.

Formalization of $FRP$ in $PV_1 + dWPHP(PV)$ was studied in [15]. We can restate the main definition of [15] in the present notation as follows.

**Definition 3.2** (in $PV_1 + dWPHP(PV)$) A *$\beta$-definable randomized algorithm* is given by a pair of $PV$-functions $\langle A, r\rangle$ such that

$$\exists w < r(\vec{x})\, A(\vec{x}, w) \neq * \rightarrow \Pr_{w<r(\vec{x})}(A(\vec{x}, w) = *) \preceq_0 \beta,$$

where $*$ is a special symbol signalling a rejecting computation, and $0 < \beta < 1$. If unspecified, we take $\beta = 1/2$.

Various properties of *FRP* were proved in $PV_1 + dWPHP(PV)$ in [15]. We will not repeat these here, but instead we will concentrate on the question of which *FRP*-algorithms are definable in $PV_1 + dWPHP(PV)$. This is actually two questions: Which *FRP*-functions are *provably* 1/2-*definable* in $PV_1 + dWPHP(PV)$, and which *TFRP*-functions are *provably total* in $PV_1 + dWPHP(PV)$. We begin with the latter.

For any *NP* search problem $S$, the statement "$S$ is total" is a $\forall\Sigma^b_1$-sentence. Conversely, for any $\forall\Sigma^b_1$-sentence $\varphi$, we can construct an *NP* search problem $S_\varphi$ such that $\varphi$ holds iff $S$ is total, thus description of provably total *NP* search problems of a theory is equivalent to characterization of its $\Sigma^b_1$-consequences. Wilkie's witnessing theorem (see [18]) states that provably total *NP* search problems of $PV_1 + dWPHP(PV)$ (or $S^1_2 + dWPHP(PV)$) are in *TFRP*, and it was shown in [15] that these witnessing *TFRP*-functions are definable and provably total in $PV_1 + dWPHP(PV)$:

**Theorem 3.3 ([15])** *Assume* $S^1_2 + dWPHP(PV) \vdash \forall x\, \exists y\, \varphi(x, y)$ *with* $\varphi \in \Sigma^b_1$*, and let* $S$ *be the corresponding search problem. There exists a probabilistic algorithm* $A$ *such that* $PV_1$ *proves*

($i$) *$A$ is 1/2-definable,*

($ii$) *$A$ solves $S$,*

*and* $PV_1 + dWPHP(PV)$ *proves that* $A$ *is total.*

It is not clear whether all *TFRP*-functions are provably total in $PV_1 + dWPHP(PV)$, or in any its r.e. extension for that matter, even if we restrict ourselves to univalued functions with values in $\{0, 1\}$, i.e., *ZPP*-predicates. On one hand, such a result cannot be shown by a relativizing technique: it would imply that *ZPP* has a complete language due to Thapen [26], and there exist oracles $A$ such that $ZPP^A$ has no complete language [5]. On the other hand, *TFRP* is widely believed to coincide with *FP*, in which case all *TFRP*-functions (but not necessarily all *TFRP*-algorithms) are trivially definable in $PV_1$.

We can obtain a more precise characterization of provably total search problems of $PV_1 + dWPHP(PV)$, if we consider "nonintensional" representations instead of particular *TFRP*-algorithms.

**Definition 3.4** A $PV$-formula $\varphi$ *represents* a search problem $S$, if the following hold (in the standard model):

($i$) if $\varphi(x, y)$, then $y$ is a solution of $S$ for $x$,

($ii$) if $x$ is a solvable instance of $S$, then $\exists y\, \varphi(x, y)$.

*WPHPWIT* is the following *NP* search problem: given a pair of circuits $G\colon 2^n \to 2^{2n}$ and $H\colon 2^{2n} \to 2^n$, find an $x < 2^{2n}$ such that $G(H(x)) \neq x$.

Let $S$ and $S'$ be *NP* search problems. $S$ is *reducible* to $S'$, if there are poly-time functions $f$ and $g$ such that:

($i$) if $x$ is a solvable instance of $S$, then $f(x)$ is a solvable instance of $S'$,

($ii$) if $y$ is a solution of $S'$ for $f(x)$, then $g(x, y)$ is a solution of $S$ for $x$.

**Theorem 3.5** *Let $S$ be an NP search problem. The following are equivalent:*

($i$) *$S$ has a provably total representation in $PV_1 + dWPHP(PV)$.*

($ii$) *$S$ is reducible to WPHPWIT.*

*Proof:* ($i$) $\to$ ($ii$) follows from Thapen's proof of Wilkie's witnessing theorem [27].

($ii$) $\to$ ($i$): assume that $S$ is given by a poly-time relation $R(x, y)$, and $f$ and $g$ form a reduction of $S$ to *WPHPWIT*. We may easily modify $f$ so that its output $f(x) = \langle G_x, H_x \rangle$ consists of a pair of circuits as in definition 3.4, provably in $PV_1 + dWPHP(PV)$. Put

$$\varphi(x, y) = R(x, y) \vee \big(G_x(H_x(y)) \neq y \wedge \neg R(x, g(x, y))\big).$$

The second disjunct never holds in the standard model by the definition of reduction, thus $\varphi$ represents $S$. $PV_1 + dWPHP(PV)$ proves $\forall x\, \exists y\, \varphi(x, y)$, as $G_x(H_x(y)) \neq y$ implies $\varphi(x, g(x, y))$ or $\varphi(x, y)$. $\square$

As noticed in [15], *WPHPWIT* can also be used as an axiomatic description of $\Sigma^b_1$-theorems of $PV_1 + dWPHP(PV)$, which is again implicit in Thapen's proof of Wilkie's witnessing theorem.

**Proposition 3.6** *The statement "WPHPWIT is total" axiomatizes $\forall\Sigma^b_1$-consequences of $PV_1 + dWPHP(PV)$ over $PV_1$.*

We return to the question which *FRP*-algorithms (not necessarily total) are definable in a given theory $T$. Perhaps surprisingly, this question is essentially equivalent to a $\forall\Sigma^b_1$-sentence, it thus reduces to the problem of the provably total *TFRP*-functions discussed above. (The constants 1/2 and 2/3 below are arbitrary.)

**Theorem 3.7** *Let $A$ be a FRP-algorithm with error 1/2. There exists a true $\forall\Sigma^b_1$-sentence $\varphi$ such that $PV_1 + dWPHP(PV)$ proves*

($i$) *if $\varphi$, then $A$ is 2/3-defined,*

($ii$) *if $A$ is 1/2-defined, then $\varphi$.*

*Moreover, the (total) NP search problem $S_\varphi$ associated with $\varphi$ is in TFRP: there exists a randomized algorithm $B$ such that $PV_1 + dWPHP(PV)$ proves*

($iii$) *if $A$ is 1/2-defined, then $B$ is 1/2-defined, total, and solves $S_\varphi$.*

*Proof:* The idea is to consider the $PV(\alpha)$-formula

$$\psi^\alpha(x, y) = \big(y < r(x) \wedge A(x, y) \neq * \to \Pr_{w<r(x)}(A(x, w) = *)_{1/50} \leq 5/8\big),$$

where $*$ is as in definition 3.2. Clearly, $HARD^A$ proves

$$\forall x, y\, \psi^\alpha(x, y) \to \ A \text{ is 2/3-defined},$$
$$A \text{ is 1/2-defined } \to \forall x, y\, \psi^\alpha(x, y).$$

We need to eliminate $\alpha$ from the formula. In the proof of theorem 2.7 (resp. lemma 2.14), the exact choice of the function $f$ is not relevant: the behaviour of $\Pr(\ldots)_{1/50}$ is preserved if we replace $\alpha$ by any average-case 1/4-hard Boolean function $f$ in the right number of variables. We thus define

$$\varphi'(x,y,f) = \big((f\colon 2^{\ell(x)} \to 2) \wedge \mathrm{Hard}^A_{1/4}(f) \to \psi^f(x,y)\big),$$

where $\ell(x) \in \mathrm{LogLog}$ is chosen as in theorem 2.7. Then $\varphi'$ is a $\Sigma^b_1$-formula, and $PV_1 + dWPHP(PV)$ proves

$$\begin{aligned} \forall x,y,f\, \varphi'(x,y,f) \to {} & A \text{ is } 2/3\text{-defined}, \\ A \text{ is } 1/2\text{-defined } & \to \forall x,y,f\, \varphi'(x,y,f). \end{aligned}$$

We use a witnessing argument to show that $S_\varphi$ is solvable in randomized polynomial time. Notice that the only non-sharply bounded existential quantifier in $\varphi'$ is the one from $\neg\,\mathrm{Hard}^A_{1/4}(f)$. $PV_1 + dWPHP(PV)$ proves the $\Sigma^b_1$-formula

$$\begin{aligned} (f,g\colon 2^{\ell(x)} \to 2) \wedge \Pr^f_{w<r(x)}(A(x,w)=*)_{1/50} > 5/8 \wedge \Pr^g_{w<r(x)}(A(x,w)=*)_{1/50} \leq 9/16 \\ \to \neg\,\mathrm{Hard}^A_{1/4}(f) \vee \neg\,\mathrm{Hard}^A_{1/4}(g). \end{aligned}$$

By Wilkie's witnessing theorem there exists a probabilistic algorithm $h(x,y,f,g) \in \mathit{TFRP}$ such that

$$\begin{aligned} (f,g\colon 2^{\ell(x)} \to 2) \wedge \Pr^f_{w<r(x)}(A(x,w)=*)_{1/50} > 5/8 \wedge \Pr^g_{w<r(x)}(A(x,w)=*)_{1/50} \leq 9/16 \\ \to \mathrm{Wit}_{\neg\,\mathrm{Hard}^A_{1/4}(f)}(h(x,y,f,g)) \vee \mathrm{Wit}_{\neg\,\mathrm{Hard}^A_{1/4}(g)}(h(x,y,f,g)) \end{aligned}$$

holds with high probability. As $A$ has error at most 1/2, the implication

$$y < r(x) \wedge A(x,y) \neq * \wedge \mathrm{Hard}^A_{1/4}(g) \to \Pr^g_{w<r(x)}(A(x,w)=*)_{1/50} \leq 9/16$$

is true. Let $B(x,y,f)$ be the probabilistic algorithm which generates a random function $g$, and applies $h(x,y,f,g)$. As most Boolean functions are average-case 1/4-hard, we have

$$\begin{aligned} y < r(x) \wedge A(x,y) \neq * \wedge (f\colon 2^{\ell(x)} \to 2) \wedge \Pr^f_{w<r(x)}(A(x,w)=*)_{1/50} > 5/8 \\ \to \mathrm{Wit}_{\neg\,\mathrm{Hard}^A_{1/4}(f)}(B(x,y,f)) \end{aligned}$$

with high probability. This construction can be easily formalized in $PV_1 + dWPHP(PV)$, using theorem 3.3 and lemma 2.2. □

### 3.2 The classes *BPP* and promise *BPP*

*BPP*, introduced by Gill [12], is arguably the most popular randomized complexity class. It is generally considered a good approximation to the class of problems which are efficiently solvable in practice.

**Definition 3.8** A language $L$ is in *BPP*, if there exists a probabilistic poly-time decision algorithm $A$ such that for every $x$,

$$x \in L \Rightarrow \Pr(A(x)) \geq 3/4,$$
$$x \notin L \Rightarrow \Pr(A(x)) \leq 1/4.$$

A *promise problem* is a pair $L = \langle L^+, L^- \rangle$ of disjoint languages. An ordinary language $L$ is identified with the promise problem $\langle L, \{0,1\}^{<\omega} \smallsetminus L \rangle$. A promise problem $L$ is in *promise BPP* ($L \in prBPP$ for short), if there exists a probabilistic poly-time algorithm $A$ such that for every $x$,

$$x \in L^+ \Rightarrow \Pr(A(x)) \geq 3/4,$$
$$x \in L^- \Rightarrow \Pr(A(x)) \leq 1/4.$$

Formalizing the definition of *prBPP* in $PV_1 + dWPHP(PV)$ is a straightforward application of the approximate counting machinery.

**Definition 3.9** (in $PV_1 + dWPHP(PV)$) Let $\beta$ be a $PV$-function with values in $(0, 1/2)$, $A$ a $PV$-predicate, and $r$ a $PV$-function. The pair $\langle A, r \rangle$ *$\beta$-defines the prBPP problem* $L_{A,r,\beta} = \langle L^+_{A,r,\beta}, L^-_{A,r,\beta} \rangle$, where

$$x \in L^+_{A,r,\beta} \quad \text{iff} \quad \Pr_{w<r(x)}(\neg A(x,w)) \preceq_0 \beta(x),$$
$$x \in L^-_{A,r,\beta} \quad \text{iff} \quad \Pr_{w<r(x)}(A(x,w)) \preceq_0 \beta(x).$$

More generally, if $L^+$, $L^-$ are disjoint definable sets, the promise problem $L = \langle L^+, L^- \rangle$ is *$\beta$-defined* by $\langle A, r \rangle$ if $L^+ \subseteq L^+_{A,r,\beta}$ and $L^- \subseteq L^-_{A,r,\beta}$.

The pair $\langle A, r \rangle$ *$\beta$-defines a BPP language* if $\forall x\, (x \in L^+_{A,r,\beta} \vee x \in L^-_{A,r,\beta})$.

If unspecified, we take $\beta = 1/4$.

**Lemma 3.10** *(in $PV_1 + dWPHP(PV)$) Let $L$ be a definable prBPP-problem, and $n \in \mathrm{Log}$. There exists a Boolean circuit $C\colon 2^n \to 2$ such that*

$$x \in L^+ \Rightarrow C(x) = 1,$$
$$x \in L^- \Rightarrow C(x) = 0,$$

*for every $x < 2^n$.*

*Proof:* Work in $HARD^A$. By lemma 2.14, there is a $PV(\alpha)$-predicate $P(x)$ such that

$$x \in L^+ \Rightarrow P(x),$$
$$x \in L^- \Rightarrow \neg P(x).$$

We may compute $P$ on a bounded interval by an oracle-free circuit, as $\alpha(x)$ only depends on the length of $x$. $\square$

**Proposition 3.11** *(in $PV_1+dWPHP(PV)$) Let $t$, $s$ be $PV$-functions such that $t(x), s(x) > 0$, and $1/s(x)+1/|t(x)| \leq 1/2$. Let $L = \langle L^+, L^- \rangle$ be a promise problem. The following are equivalent.*

*(i) $L$ is a $(1/2 - 1/|t|)$-definable prBPP-problem,*

*(ii) $L$ is a $1/4$-definable prBPP-problem,*

*(iii) $L$ is a $1/s$-definable prBPP-problem.*

*Proof:* The only interesting implication is $(i) \to (iii)$. Assume that $L$ is $(1/2 - 1/|t|)$-defined by $\langle A, r \rangle$. Let $c$ be the constant from proposition 2.18, put $m(x) = |t(x)|^2|cs(x)|$, $r'(x) = r(x)^{m(x)}$, and

$$A'(x, w') \leftrightarrow \big(|\{i < m(x) \mid A(x, w_i)\}| \geq m(x)/2\big),$$

where $w' < r'(x)$ is viewed as a sequence $\langle w_i \mid i < m(x) \rangle$ of numbers less than $r(x)$. Then $L$ is $1/s$-defined by $\langle A', r' \rangle$ due to Chernoff's bound (proposition 2.18). □

Notice that *prBPP* is defined by a purely syntactic condition: in other words, every pair $\langle A, r \rangle$ of $PV$-functions (provably) defines a *prBPP*-problem.

**Corollary 3.12** *Every prBPP-algorithm is definable in $PV_1 + dWPHP(PV)$.*

Definable *BPP*-languages are essentially "provably total" *prBPP*-problems. As in the case of *TFRP*, we do not know whether all *BPP*-languages are definable in $PV_1 + dWPHP(PV)$ or its r.e. extension; again, relativizing techniques cannot work, as Thapen's result is applicable to *BPP*, and an oracle with respect to which *BPP* does not have a complete language was constructed in [14]. We show that the totality of a *BPP*-algorithm is essentially equivalent to a $\forall\Sigma^b_1$-sentence, thus the characterization of the *BPP*-languages definable in a particular theory can be reduced to the characterization of its provably total *TFRP*-functions.

**Theorem 3.13** *Let $A$ be a BPP-algorithm. There exists a true $\forall\Sigma^b_1$-sentence $\varphi$ such that $PV_1 + dWPHP(PV)$ proves*

*(i) if $\varphi$, then $A$ $1/3$-defines a BPP-language,*

*(ii) if $A$ $1/4$-defines BPP-language, then $\varphi$.*

*Moreover, the NP search problem $S_\varphi$ associated with $\varphi$ is in TFRP. There is a randomized algorithm $B$ such that $PV_1 + dWPHP(PV)$ proves*

*(iii) if $A$ $1/4$-defines BPP-language, then $B$ is $1/2$-defined, total, and solves $S_\varphi$.*

*Proof:* We define

$$\varphi = \forall x \, \forall f \,\big((f \colon 2^{\ell(x)} \to 2) \wedge \mathrm{Hard}^A_{1/4}(f) \\ \to \Pr^f_{w<r(x)}(A(x,w))_{1/50} < 7/24 \vee \Pr^f_{w<r(x)}(\neg A(x,w))_{1/50} < 7/24\big)$$

with suitably chosen $\ell(x) \in \mathrm{LogLog}$, and proceed as in the proof of 3.7.

There is a minor complication in the construction of the probabilistic solver to $S_\varphi$: the algorithm cannot directly decide which of the disjuncts in $\varphi$ should hold, as we do not know whether $BPP = ZPP$. The solution is to try both possibilities, and check whether either of them leads to a correct witness for $\neg\,\mathrm{Hard}^A_{1/4}(f)$. $\square$

A similar argument can be used to prove that *prBPP* lies on the second level of the polynomial hierarchy. The original result (formulated for *BPP* only) is due to Sipser and Gács [25], and it was simplified by Lautemann [20]. We follow an alternative proof due to Nisan and Wigderson [22].

**Proposition 3.14** *Let $A$ be a $PV$-predicate, and $r$ a $PV$-functions. There are $\Sigma^b_2$-formulas $\sigma^+(x), \sigma^-(x)$ and $\Pi^b_2$-formulas $\pi^+(x), \pi^-(x)$ such that $PV_1 + dWPHP(PV)$ proves*

$$x \in L^+_{A,r,1/4} \to \pi^+(x) \to \sigma^+(x) \to x \in L^+_{A,r,1/3},$$
$$x \in L^-_{A,r,1/4} \to \pi^-(x) \to \sigma^-(x) \to x \in L^-_{A,r,1/3}.$$

*In particular, any definable BPP-language is in $\Sigma^b_2 \cap \Pi^b_2$.*

*Proof:* It suffices to define

$$\pi^+(x) = \forall f\,\big(f\colon 2^{\ell(x)} \to 2 \wedge \mathrm{Hard}^A_{1/4}(f) \to \Pr^f_{w<r(x)}(\neg A(x,w))_{1/50} \le 7/24\big),$$
$$\sigma^+(x) = \exists f\,\big(f\colon 2^{\ell(x)} \to 2 \wedge \mathrm{Hard}^A_{1/4}(f) \wedge \Pr^f_{w<r(x)}(\neg A(x,w))_{1/50} \le 7/24\big),$$
$$\pi^-(x) = \forall f\,\big(f\colon 2^{\ell(x)} \to 2 \wedge \mathrm{Hard}^A_{1/4}(f) \to \Pr^f_{w<r(x)}(A(x,w))_{1/50} \le 7/24\big),$$
$$\sigma^-(x) = \exists f\,\big(f\colon 2^{\ell(x)} \to 2 \wedge \mathrm{Hard}^A_{1/4}(f) \wedge \Pr^f_{w<r(x)}(A(x,w))_{1/50} \le 7/24\big).$$

The quantifiers over $f$ are bounded as $f \le 2^{2^{\ell(x)}}$ and $\ell(x) = O(||x||)$. $\square$

To complete the picture we mention an elegant alternative description of definable *BPP*-languages, based on implicit definability in (extensions of) $HARD^A$. The intuition behind this characterization stems from the well-known result $BPP = \textit{almost-}P$ (cf. [3, 22]).

**Definition 3.15** Let $T$ be a simple extension of $PV_1 + dWPHP(PV)$, and $T^+(\alpha) := T + HARD^A$. A $PV(\alpha)$-predicate $P^\alpha(x)$ is a $T^+$*-definable implicitly poly-time predicate*, if

$$T^+(\alpha) + T^+(\beta) \vdash P^\alpha(x) \leftrightarrow P^\beta(x).$$

**Theorem 3.16** *Let $T$ be a simple extension of $PV_1 + dWPHP(PV)$.*

*(i) Every $T$-provably total BPP-language is in $T^+$ equivalent to a $T^+$-definable implicitly poly-time predicate.*

*(ii) Every $T^+$-definable implicitly poly-time predicate is in $T^+$ equivalent to a $T$-provably total BPP-language.*

*Proof:* $(i)$: let $L$ be a definable *BPP*-language. By lemma 2.14, there exists a $PV(\alpha)$-predicate $P^\alpha$ such that

$$T^+(\alpha) \vdash P^\alpha(x) \leftrightarrow x \in L.$$

Then clearly

$$T^+(\alpha), T^+(\beta) \vdash P^\alpha(x) \leftrightarrow P^\beta(x).$$

$(ii)$: assume that

$$T^+(\alpha), T^+(\beta) \vdash P^\alpha(x) \leftrightarrow P^\beta(x).$$

Let $c$ be a constant such that $P^\alpha(x)$ only accesses the value of $\alpha(y)$ for $||y|| \leq c||x||$. Work in $T^+(\alpha)$. Fix $x$, let $f = \langle f_i \mid i \leq c||x|| \rangle$ be a sequence of average-case 1/4-hard functions $f_i \colon 2^i \to 2$, and define

$$\beta(y) = \begin{cases} f_{||y||}, & ||y|| \leq c||x||, \\ \alpha(y), & \text{otherwise.} \end{cases}$$

Then $\beta$ defines a (parametric) interpretation of $T^+(\beta)$ in $T^+(\alpha)$, and consequently $P^\alpha(x) \leftrightarrow P^f(x)$.

We thus have

$$T^+(\alpha) \vdash \forall i \leq c||x|| \left( f_i \colon 2^i \to 2 \wedge \mathrm{Hard}^A_{1/4}(f_i) \right) \to (P^\alpha(x) \leftrightarrow P^f(x)).$$

Let $A$ be the formalization of the following randomized algorithm: on input $x$, generate a random sequence $f = \langle f_i \mid i \leq c||x|| \rangle$ of functions $f_i \colon 2^i \to 2$, and output $P^f(x)$. By lemma 4.10 in [15], $PV_1 + dWPHP(PV)$ proves

$$Pr_f\left(\neg \forall i \leq c||x|| \ \mathrm{Hard}^A_{1/4}(f_i)\right) \preceq_0 1/4,$$

thus

$$T^+(\alpha) \vdash \Pr_f\left(A(x,f) \leftrightarrow \neg P^\alpha(x)\right) \preceq_0 1/4.$$

In particular,

$$T^+(\alpha) \vdash \Pr_f(A(x,f)) \preceq_0 1/4 \vee \Pr_f(\neg A(x,f)) \preceq_0 1/4,$$

i.e., $A$ is a 1/4-defined *BPP*-algorithm in $T^+(\alpha)$, and by theorem 2.13, also in $T$. If $L$ denotes the *BPP*-language defined by $A$, clearly

$$T^+(\alpha) \vdash P^\alpha(x) \leftrightarrow x \in L$$

as required. □

### 3.3 The class *APP*

The class *APP* is a generalization of *BPP* introduced by Kabanets, Rackoff, and Cook [17]. It comprises a representative class of algorithms which can be derandomized using the current methods for proving $P = BPP$ (viz. hardness-randomness tradeoffs), and unlike *BPP*, it is known to have a complete problem. A unique feature of *APP* is that it does not consist of languages (or promise problems), but functions with real values in the interval $[0, 1]$.

**Definition 3.17** A real-valued function $f\colon \omega \to [0,1]$ is in $APP$, if there exists a probabilistic poly-time function $g(x,y)$ with values in $[0,1]_{\mathbb{Q}}$ such that

$$\Pr\bigl(|f(x) - g(x,2^k)| \le 1/k\bigr) \ge 3/4$$

for all $x$ and $k$.

We cannot directly talk about real numbers in bounded arithmetic, we thus have to formalize $APP$-algorithms without an explicit reference to the functions which they compute. The idea is similar to methods used in constructive analysis (cf. [4]).

**Definition 3.18** (in $PV_1 + dWPHP(PV)$) Let $\beta(x,y)$ be a $PV$-function with rational values in $(0,1/2)$. A *$\beta$-definable $APP$-algorithm* is given by a pair of $PV$-functions $g(x,y,w)$ and $r(x,y)$, where $r$ has positive integer values, $g$ has rational values in $[0,1]$, and

$$\begin{aligned}\forall x\,\forall k,\ell \in \mathrm{Log}\;\exists a \in [0,1]\,\bigl(&\Pr_{w<r(x,2^k)}\bigl(|g(x,2^k,w) - a| > 1/k\bigr) \preceq_0 \beta(x,2^k)\\ &\wedge \Pr_{w<r(x,2^\ell)}\bigl(|g(x,2^\ell,w) - a| > 1/\ell\bigr) \preceq_0 \beta(x,2^\ell)\bigr).\end{aligned}$$

When unspecified, we take $\beta = 1/4$.

Let $\langle g', r'\rangle$ be a $\beta'$-definable $APP$-algorithm. We say that $\langle g,r\rangle$ and $\langle g',r'\rangle$ *compute the same function* if

$$\begin{aligned}\forall x\,\forall k \in \mathrm{Log}\;\exists a \in [0,1]\,\bigl(&\Pr_{w<r(x,2^k)}\bigl(|g(x,2^k,w) - a| > 1/k\bigr) \preceq_0 \beta(x,2^k)\\ &\wedge \Pr_{w<r'(x,2^k)}\bigl(|g'(x,2^k,w) - a| > 1/k\bigr) \preceq_0 \beta'(x,2^k)\bigr).\end{aligned}$$

**Proposition 3.19** *(in $PV_1 + dWPHP(PV)$) Let $t(x,y)$ and $s(x,y)$ be $PV$-functions with positive integer values. If $\langle g,r\rangle$ is a $(1/2 - 1/|t|)$-definable $APP$-algorithm, there exists a $1/s$-definable $APP$-algorithm $\langle g',r'\rangle$ which computes the same function as $\langle g,r\rangle$.*

*Proof:* Let $c$ be the constant from proposition 2.18, and let $m(x,y) := |cs(x,y)||t(x,y)|^2 \in \mathrm{Log}$. Put

$$\begin{aligned} r'(x,y) &= r(x,y)^{m(x,y)},\\ g'(x,y,w') &= \mathrm{Median}(g(x,y,w_0),\dots,g(x,y,w_{m-1})),\end{aligned}$$

where $w' < r'(x,y)$ is considered as a sequence of $m = m(x,y)$ numbers $w_i < r(x,y)$. Fix $x$, $k \in \mathrm{Log}$, and $a \in [0,1]$ such that

$$\Pr_{w<r}\bigl(|g(x,2^k,w) - a| > 1/k\bigr) \preceq_0 1/2 - 1/|t|.$$

By proposition 2.18 (Chernoff's bound), we have

$$\Pr_{w'<r'}\Bigl(\bigl|\bigl\{i < m \mid |g(x,2^k,w_i) - a| > 1/k\bigr\}\bigr| \ge m/2\Bigr) \preceq_0 c4^{-m/|t|^2} \le 1/s.$$

The median of a set of numbers falls into the interval $I = [a - 1/k, a + 1/k]$ whenever more than half of the numbers are in $I$, thus

$$\Pr_{w'<r'}\bigl(|g'(x,2^k,w') - a| > 1/k\bigr) \preceq_0 1/s. \qquad \square$$

**Definition 3.20** An $APP$-function $f\colon \omega \to [0,1]$ is *representable in a theory* $T$, if there exists a pair of $PV$-functions $\langle g, r\rangle$ which, provably in $T$, 1/4-defines an $APP$-algorithm, and for any $x$ and $k$,

$$\Pr_{w<r(x,2^k)}\big(|g(x,2^k,w) - f(x)| > 1/k\big) \leq 1/4$$

is true in $\mathbb{N}$.

We want to show that all $APP$-functions are representable in $PV_1 + dWPHP(PV)$. Notice that for any reasonable model of computation (such as $APP$), the class of algorithms representable in a given recursively axiomatizable theory is recursively enumerable. We thus need to establish recursive enumerability of $APP$ as a necessary prerequisite (it was left as an open problem in [17]).

**Definition 3.21** Let $f, g\colon \omega \to [0,1]$ be real-valued functions. We say that $f$ is (*poly-time many-one approximately*) *reducible* to $g$, if there is a poly-time function $r$ such that for every $x$ and $k$,

$$|f(x) - g(r(x,2^k))| \leq 1/k.$$

The *Circuit Acceptance Probability Problem* ($CAPP$) is the real-valued function $f_{CAPP}$ such that for every Boolean circuit $C\colon 2^n \to 2$,

$$f_{CAPP}(C) = \Pr_{u<2^n}(C(u) = 1).$$

**Theorem 3.22 ([17])** *A function $f$ is in $APP$ if and only if $f$ is reducible to $f_{CAPP}$.*

**Theorem 3.23** *The class $APP$ is recursively enumerable. I.e., there exists a recursive sequence $\{A_e \mid e \in \omega\}$ such that*

- *each $A_e$ is a description of an $APP$-algorithm approximating a function $f_e$,*
- *for every $f \in APP$, there is an $e$ such that $f = f_e$.*

*Proof:* Let $\{g_e \mid e \in \omega\}$ be a recursive enumeration of all clocked poly-time algorithms $g(x,y)$, such that the output of $g(x,y)$ is a description of a Boolean circuit. Let $\mathrm{Cut}_p^q$ be the cut-off function

$$\mathrm{Cut}_p^q(x) := \max(p, \min(q,x)) = \begin{cases} q, & x \geq q, \\ x, & p \leq x \leq q, \\ p, & x \leq p. \end{cases}$$

Let $A_e(x,2^k)$ be the algorithm described in figure 1. Clearly, $A_e$ is a probabilistic poly-time algorithm. Fix $e$ and $x$, and define

$$\begin{aligned} C_i &:= g_e(x,2^i), \\ a_i &:= \Pr_u(C_i(u) = 1), \\ b_k &:= \mathrm{Cut}_0^1\Big(a_1 + \sum_{i=2}^{k} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(a_i - a_{i-1})\Big). \end{aligned}$$

| |
|---|
| input: $x$, $2^k$<br>for $i = 1, \dots, k$ do:<br>   $C_i \leftarrow g_e(x, 2^i)$<br>   whp, compute $c_i$ such that $\lvert c_i - \Pr_u(C_i(u) = 1)\rvert \le 1/(4k^2)$ by random sampling<br>output $\mathrm{Cut}_0^1\Big(c_1 + \sum_{i=2}^{k} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(c_i - c_{i-1})\Big)$ |

Figure 1: the $APP$-algorithm $A_e$

For any $k < \ell$, we have

$$|b_\ell - b_k| \le \Big|\sum_{i=k+1}^{\ell} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(a_i - a_{i-1})\Big| \le \sum_{i=k+1}^{\ell} \frac{1}{2i^2} \le \frac{1}{2}\sum_{i=k+1}^{\infty} \frac{1}{(i-1)i} = \frac{1}{2k},$$

thus the sequence $\{b_k \mid k \in \omega\}$ is Cauchy, and converges to a number $f_e(x) := b \in [0,1]$ such that

$$|b - b_k| = \lim_{\ell\to\infty} |b_\ell - b_k| \le \frac{1}{2k}.$$

Fix $k$, and consider a computation of $A_e$ on input $\langle x, 2^k\rangle$. For all $i = 1, \dots, k$, let $c_i \in [0,1]$ be as in figure 1. With high probability, we have

$$|c_i - a_i| \le \frac{1}{4k^2}$$

for every $i$. Let

$$d := \mathrm{Cut}_0^1\Big(c_1 + \sum_{i=2}^{k} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(c_i - c_{i-1})\Big)$$

be the output of the algorithm. Addition, subtraction, and the cut-off function are 1-Lipschitz, thus

$$\begin{aligned}
|d - b_k| &\le |c_1 - a_1| + \sum_{i=2}^{k} |(c_i - c_{i-1}) - (a_i - a_{i-1})| \\
&\le |c_1 - a_1| + \sum_{i=2}^{k} \big(|c_i - a_i| + |c_{i-1} - a_{i-1}|\big) \\
&\le (2k-1)\frac{1}{4k^2} \le \frac{1}{2k},
\end{aligned}$$

and

$$|d - b| \le |d - b_k| + |b_k - b| \le \frac{1}{k}.$$

This means that $A_e$ is an $APP$-algorithm for $f_e$.

Let $f$ be an arbitrary $APP$-function. By $APP$-completeness of CAPP, there is a poly-time function $g$ such that for any $x$ and $k$, $C := g(x, 2^k)$ is a Boolean circuit satisfying

$$|f(x) - \Pr_u(C(u) = 1)| \le \frac{1}{k}.$$

Choose $e$ such that

$$g_e(x, 2^k) = g(x, 2^{4(k+1)^2}).$$

Fix $x$, and define the sequences $C_i$, $a_i$, and $b_i$ as above. We have

$$|a_i - a_{i-1}| \leq \frac{1}{4(i+1)^2} + \frac{1}{4i^2} \leq \frac{1}{2i^2},$$

thus

$$b_k = \mathrm{Cut}_0^1\Big(a_1 + \sum_{i=2}^{k}(a_i - a_{i-1})\Big) = \mathrm{Cut}_0^1(a_k) = a_k,$$

which means

$$f_e(x) = \lim_{k\to\infty} b_k = \lim_{k\to\infty} a_k = f(x).$$

As $x$ was arbitrary, $f_e = f$. □

**Lemma 3.24** *CAPP is representable in $PV_1 + dWPHP(PV)$.*

*Proof:* Let $\langle g, r\rangle$ be the formalization of the following algorithm: given $C$ and $2^k$, choose a random Boolean function $f$ in a suitable number of variables, and output $\Pr_u^f(C(u) = 1)_{1/(3k)}$.

Fix $C\colon 2^n \to 2$, $k < \ell \in \mathrm{Log}$, and let $\xi = 1/(3\ell)$, $\Pr_u(C(u) = 1) \approx_\xi a$. As $PV_1 + dWPHP(PV)$ proves

$$\Pr_f(\neg\,\mathrm{Hard}^A_{1/4}(f)) \preceq_0 1/4$$

(lemma 2.2), we have

$$\Pr_f\big(|g(C, 2^k, f) - a| > 1/(3k) + \xi + \xi\big) \preceq_0 1/4$$

and

$$\Pr_f\big(|g(C, 2^\ell, f) - a| > 1/(3\ell) + \xi + \xi\big) \preceq_0 1/4$$

by lemma 2.11 *(ii)*. □

We remark that the combinatorial core of theorem 3.22 can also be formalized in $PV_1 + dWPHP(PV)$ with no difficulty. However, we do not know how to sensibly *formulate* the statement of theorem 3.22 in $PV_1 + dWPHP(PV)$, due to absence of real numbers in bounded arithmetic.

**Theorem 3.25** *Every APP-function $f$ is representable in $PV_1 + dWPHP(PV)$.*

*Proof:* The basic idea is to partially formalize theorem 3.23 in $PV_1 + dWPHP(PV)$.

As in theorem 3.23, choose a $PV$-function $h(x, y)$ such that for every $x$ and $k$ we have

$$|f(x) - \Pr_u(C(u) = 1)| \leq 1/(8k^2),$$

where $C = h(x, 2^k)$. Let $\langle g_{CAPP}, r_{CAPP}\rangle$ be the representation of CAPP from lemma 3.24, amplified by proposition 3.19 so that the error on input $\langle C, 2^k\rangle$ is at most $1/k$. We may assume that $r_{CAPP}(x, 2^k)$ is always a power of 2. Define $g(x, 2^k, w)$ as in figure 2, and let $r(x, 2^k)$ be a power of 2 large enough to accommodate all calls to $g_{CAPP}$ inside $g$. The functions $\langle g, r\rangle$

| |
|---|
| input: $x$, $2^k$, $w$<br>for $i = 1, \dots, k$ do:<br>  $C_i \leftarrow h(x, 2^i)$<br>  $c_i \leftarrow g_{CAPP}\big(C_i, 2^{1/(8k^2)}, \big(w \bmod r_{CAPP}(C_i, 2^{1/(8k^2)})\big)\big)$<br>output $\mathrm{Cut}_0^1\Big(c_1 + \sum_{i=2}^{k} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(c_i - c_{i-1})\Big)$ |

Figure 2: the function $g$, formalizing $A_e$

represent $f$ by the proof of theorem 3.23, it remains to prove in $PV_1 + dWPHP(PV)$ that $\langle g, r\rangle$ is a 1/4-defined $APP$-algorithm.

Work in $HARD^A$. Fix $x$, and $k < \ell \in \mathrm{Log}$. Define

$$\begin{aligned} C_i &:= h(x, 2^i), \\ a_i &:= \Pr_u(C_i(u) = 1)_{1/(10\ell^2)}, \\ a &:= \mathrm{Cut}_0^1\Big(a_1 + \sum_{i=2}^{\ell} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(a_i - a_{i-1})\Big), \\ a' &:= \mathrm{Cut}_0^1\Big(a_1 + \sum_{i=2}^{k} \mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(a_i - a_{i-1})\Big) \end{aligned}$$

for every $i \le \ell$. Consider first the computation of $d := g(x, 2^\ell, w)$ on a random input $w$, and let $c_i$ be as in figure 2. For every $i \le \ell$ and suitably chosen small $\xi$, we have

$$|c_i - a_i| \le \frac{1}{8\ell^2} + \frac{1}{10\ell^2} + \xi \le \frac{1}{4\ell^2}$$

with probability at least $1 - 1/(8\ell^2)$, thus

$$\forall i \le \ell\, |c_i - a_i| \le 1/(4\ell^2)$$

with probability $1 - \ell/(8\ell^2) - \xi \ge 3/4$ by proposition 2.15. When this happens, we have

$$|d - a| \le |c_1 - a_1| + \sum_{i=2}^{\ell}\big(|c_i - a_i| + |c_{i-1} - a_{i-1}|\big) \le \frac{2\ell - 1}{4\ell^2} < \frac{1}{2\ell}$$

as in theorem 3.23, thus

$$\Pr_w\big(|g(x, 2^\ell, w) - a| > 1/\ell\big) \preceq_0 1/4.$$

Now consider the computation of $d' := g(x, 2^k, w)$. We have

$$|d' - a'| < \frac{1}{2k}$$

with probability at least 3/4 by the same reasoning as above. Moreover,

$$|a' - a| \le \sum_{i=k+1}^{\ell} \big|\mathrm{Cut}_{-1/(2i^2)}^{1/(2i^2)}(a_i - a_{i-1})\big| \le \sum_{i=k+1}^{\ell} \frac{1}{2(i-1)i} = \frac{1}{2}\left(\frac{1}{k} - \frac{1}{\ell}\right),$$

where the last equality follows by induction on $\ell$. Consequently

$$|g(x, 2^k, w) - a| \leq \frac{1}{2k} + \frac{1}{2k} - \frac{1}{2\ell} < \frac{1}{k}$$

holds with probability at least 3/4. □

### 3.4 The classes *MA* and promise *MA*

Babai [1] (cf. [2]) introduced a hierarchy of complexity classes based on public-coin randomized interactive proof systems, *Arthur-Merlin games*. The game is played by the omniscient but untrustworthy wizard Merlin, and king Arthur, who may flip coins, but otherwise his computational power is polynomially limited. The players exchange messages in turn, and the goal for Merlin is to convince mistrustful Arthur to accept the input string. *MA* is the lowest level of the hierarchy, where the game is restricted to one round, with Merlin playing first.

**Definition 3.26** A promise problem $L$ is in *promise MA* (*prMA* for short), if there exists a probabilistic poly-time algorithm $A(x, y)$ such that

$$A(x, y) \Rightarrow |y| \leq p(|x|)$$

for some polynomial $p$, and

$$x \in L^+ \Rightarrow \exists y \ \Pr(A(x, y)) \geq 3/4,$$
$$x \in L^- \Rightarrow \forall y \ \Pr(A(x, y)) \leq 1/4.$$

A language is in *MA* if the corresponding promise problem is in *prMA*.

**Definition 3.27** (in $PV_1 + dWPHP(PV)$) Let $\beta$ be a $PV$-function with values in $(0, 1/2)$, $A$ a $PV$-predicate, and $q, r$ $PV$-functions. The triple $\langle A, q, r\rangle$ *$\beta$-defines a prMA-problem* $L = \langle L^+, L^-\rangle$ if $L^+ \supseteq L^{+\exists}_{A,q,r,\beta}$ and $L^- \supseteq L^{-\forall}_{A,q,r,\beta}$, where

$$x \in L^{+\exists}_{A,q,r,\beta} \quad \text{iff} \quad \exists y \leq q(x) \ \Pr_{w<r(x)}(\neg A(x, y, w)) \preceq_0 \beta(x),$$
$$x \in L^{-\forall}_{A,q,r,\beta} \quad \text{iff} \quad \forall y \leq q(x) \ \Pr_{w<r(x)}(A(x, y, w)) \preceq_0 \beta(x).$$

$\langle A, r, s\rangle$ $\beta$-defines an $MA$-language, if $\forall x\, (x \in L^{+\exists}_{A,r,s,\beta} \vee x \in L^{-\forall}_{A,r,s,\beta})$. If unspecified, we take $\beta = 1/4$.

**Corollary 3.28** (*in* $PV_1 + dWPHP(PV)$) *Let $t$ and $s$ be as in proposition* 3.11*, and let $L = \langle L^+, L^-\rangle$ be a promise problem. The following are equivalent.*

(*i*) *$L$ is a $(1/2 - 1/|t|)$-definable prMA-problem,*

(*ii*) *$L$ is a 1/4-definable prMA-problem,*

(*iii*) *$L$ is a $1/s$-definable prMA-problem.*

*Moreover, every definable prMA-problem is in* (*the natural formalization of*) *prNP/poly.*

*Proof:* This follows from proposition 3.11 and lemma 3.10, as the definable *prMA*-problems are just existentially quantified definable *prBPP*-problems. □

Trivially, every *prMA*-problem is representable in $PV_1 + dWPHP(PV)$. For *MA*-languages, we again have a reduction to a $\Sigma^b_1$-problem.

**Proposition 3.29** *Let $A$ be an MA-algorithm. There exists a true $\forall\Sigma^b_1$-sentence $\varphi$ such that $PV_1 + dWPHP(PV)$ proves*

(*i*) *if $\varphi$, then $A$ $1/3$-defines an MA-language,*

(*ii*) *if $A$ $1/4$-defines an MA-language, then $\varphi$.*

*Proof:* We may take the formula

$$\varphi = \forall x\,\forall f\,\forall y\,\exists z\,\big(\neg\,\mathrm{Hard}^A_{1/4}(f) \vee \Pr^f_w(A(x,y,w))_{1/50} \le 7/24 \\ \vee \Pr^f_w(\neg A(x,z,w))_{1/50} \le 7/24\big)$$

as in theorem 3.13. □

We do not know whether the *NP* search problem associated with $\varphi$ is solvable in probabilistic polynomial time. It holds at least for languages from $NP^{BPP} \subseteq MA$.

**Proposition 3.30** *Let $A, q$ and $r$ be PV-functions. There are $\Sigma^b_2$-formulas $\sigma^+(x), \sigma^-(x)$ and $\Pi^b_2$-formulas $\pi^+(x), \pi^-(x)$ such that $PV_1 + dWPHP(PV)$ proves*

$$x \in L^{+\exists}_{A,q,r,1/4} \to \pi^+(x) \to \sigma^+(x) \to x \in L^{+\exists}_{A,q,r,1/3},$$
$$x \in L^{-\forall}_{A,q,r,1/4} \to \pi^-(x) \to \sigma^-(x) \to x \in L^{-\forall}_{A,q,r,1/3}.$$

*In particular, any definable MA-language is in $\Sigma^b_2 \cap \Pi^b_2$.*

*Proof:* Similar to proposition 3.14. The extra quantifiers do no harm:

$$\pi^+(x) = \forall f\,\big(\neg\,\mathrm{Hard}^A_{1/4}(f) \vee \exists y \le q(x)\ \Pr^f_{w<r(x)}(\neg A(x,y,w))_{1/50} \le 7/24\big),$$
$$\sigma^+(x) = \exists f\,\exists y \le q(x)\,\big(\mathrm{Hard}^A_{1/4}(f) \wedge \Pr^f_{w<r(x)}(\neg A(x,y,w))_{1/50} \le 7/24\big),$$
$$\pi^-(x) = \forall f\,\forall y \le q(x)\,\big(\neg\,\mathrm{Hard}^A_{1/4}(f) \vee \Pr^f_{w<r(x)}(A(x,y,w))_{1/50} \le 7/24\big),$$
$$\sigma^-(x) = \exists f\,\big(\mathrm{Hard}^A_{1/4}(f) \wedge \forall y \le q(x)\ \Pr^f_{w<r(x)}(A(x,y,w))_{1/50} \le 7/24\big),$$

where $f$ is bounded as in proposition 3.14. □

# 4 Relativization and $AM$

The content of section 2 can be relativized in a straightforward way: we work with $PV(R)$ instead of $PV$, where $R$ is a new predicate, and we replace circuits with oracle circuits. The relativized version of theorem 2.7 then provides approximate counting of sets defined by oracle circuits in $PV_1(R) + dWPHP(PV(R))$. The other results relativize in a similar way.

In particular, counting of sets higher in the polynomial hierarchy may be achieved by substitution of $\Sigma^b_i$-predicates for $R$. Namely, approximate counting of $P^{\Sigma^b_i}$-definable sets (or more generally, sets defined by circuits with $\Sigma^b_i$ oracles) is possible in $T^i_2 + dWPHP(FP^{\Sigma^b_i}) \subseteq T^{i+2}_2$. Relativization of section 3 provides the formalization of $FRP^{\Sigma^b_i}$, $prBPP^{\Sigma^b_i}$, $APP^{\Sigma^b_i}$, and $prMA^{\Sigma^b_i}$ in $T^i_2 + dWPHP(FP^{\Sigma^b_i})$.

Approximate counting of $NP$ sets also permits formalization of Babai's class $AM$ [1], which is defined by one-round Arthur-Merlin games where Arthur plays first.

**Definition 4.1** A promise problem $L$ is in *promise* $AM$ ($prAM$ for short), if there exists a probabilistic poly-time algorithm $A(x, y)$ such that

$$A(x, y) \Rightarrow |y| \le p(|x|)$$

for some polynomial $p$, and

$$\begin{aligned} x \in L^+ &\Rightarrow \Pr(\exists y\, A(x, y)) \ge 3/4, \\ x \in L^- &\Rightarrow \Pr(\exists y\, A(x, y)) \le 1/4. \end{aligned}$$

A language is in $AM$ if the corresponding promise problem is in $prAM$.

**Definition 4.2** (in $T^1_2 + dWPHP(FP^{\Sigma^b_1})$) Let $\beta$ be a $PV$-function with values in $(0, 1/2)$. A pair $\langle\varphi, r\rangle$, where $\varphi(x, w)$ is a $\Sigma^b_1$-formula, and $r$ is a $PV$-function, *$\beta$-defines a $prAM$ problem* $L = \langle L^+, L^-\rangle$ if $L^+ \supseteq L^+_{\varphi,r,\beta}$ and $L^- \supseteq L^-_{\varphi,r,\beta}$, where

$$\begin{aligned} x \in L^+_{\varphi,r,\beta} \quad &\text{iff} \quad \Pr_{w<r(x)}(\neg\varphi(x, w)) \preceq^1_0 \beta(x), \\ x \in L^-_{\varphi,r,\beta} \quad &\text{iff} \quad \Pr_{w<r(x)}(\varphi(x, w)) \preceq^1_0 \beta(x), \end{aligned}$$

and $\preceq^i_\varepsilon$ denotes $\preceq_\varepsilon$ relativized with a $\Sigma^b_i$-complete oracle. The pair $\langle\varphi, r\rangle$ *$\beta$-defines an $AM$-language,* if $\forall x\, (x \in L^+_{\varphi,r,\beta} \vee x \in L^+_{\varphi,r,\beta})$.

If unspecified, we take $\beta = 1/4$.

**Proposition 4.3** *(in $T^1_2 + dWPHP(FP^{\Sigma^b_1})$) Let $t$ and $s$ be as in proposition* 3.11*, and let $L = \langle L^+, L^-\rangle$ be a promise problem. The following are equivalent:*

*(i) $L$ is a $(1/2 - 1/|t|)$-definable $prAM$-problem,*

*(ii) $L$ is a $1/4$-definable $prAM$-problem,*

*(iii) $L$ is a $1/s$-definable $prAM$-problem.*

*Proof:* As $prAM \subseteq prBPP^{NP}$, the result follows from relativization of proposition 3.11, observing that the formula $\varphi'(x, w)$ defined by

$$\big|\{i < m(x) \mid \varphi(x, w_i)\}\big| \ge m(x)/2$$

is $\Sigma^b_1$, as it is equivalent to

$$\exists I \subseteq m(x)\, \big(|I| \ge m(x)/2 \wedge \forall i \in I\, \varphi(x, w_i)\big). \qquad \square$$

Babai's Collapse Theorem [1] states that $AM$ coincides with the class of languages recognized by an Arthur-Merlin protocol with a bounded number of rounds. It is not clear how to define general Arthur-Merlin games in bounded arithmetic; the next theorem shows that $prMAM = prAM$, which implies that any class obtained by a constant number of applications of the $\exists$ and $BP$ operators to $prP$ is contained in $prAM$.

**Theorem 4.4** *(in $T^1_2 + dWPHP(FP^{\Sigma^b_1})$) Let $L = \langle L^+, L^- \rangle$ be a $1/4$-definable $prAM$-problem, and $q$ a $PV$-function. Define a promise problem $L^\exists = \langle L^{+\exists}, L^{-\forall} \rangle$ by*

$$\begin{aligned} x \in L^{+\exists} \quad &\textit{iff} \quad \exists y < q(x)\, \langle x, y \rangle \in L^+, \\ x \in L^{-\forall} \quad &\textit{iff} \quad \forall y < q(x)\, \langle x, y \rangle \in L^-. \end{aligned}$$

*Then $L^\exists$ is a $1/4$-definable $prAM$-problem.*

*In particular, every definable prMA-problem is a definable $prAM$-problem.*

*Proof:* By proposition 4.3, there exists a $1/(4q(x))$-definition $\langle \varphi, r \rangle$ of $L$. Define

$$\varphi'(x, w) \quad \text{iff} \quad \exists y < q(x)\, \varphi(x, y, w).$$

Then $\langle \varphi', r \rangle$ is a $1/4$-definition of $L^\exists$: if $x \in L^{+\exists}$, there exists a $y < q(x)$ such that $\Pr_w(\neg\varphi(x, y, w)) \preceq^1_0 1/(4q(x))$, and a fortiori

$$\Pr_w(\neg\exists y < q(x)\, \varphi(x, y, w)) \preceq^1_0 1/(4q(x)) \le 1/4.$$

Assume $x \in L^{-\forall}$. Then $\Pr_w(\varphi(x, y, w)) \preceq^1_0 1/(4q(x))$ for every $y < q(x)$, and we would like to argue that

$$\Pr_w(\exists y < q(x)\, \varphi(x, y, w)) \preceq^1_0 1/4.$$

We cannot do it directly (say, by application of proposition 2.19), as $q(x) \notin \mathrm{Log}$ in general, but we can explore the fact that the proof of proposition 4.3 is sufficiently uniform.

We work in the relativized version of $HARD^A$, which we denote $HARD^A(\Sigma^b_1)$. Let $\langle \psi, s \rangle$ be a $1/6$-definition of $L$, and we assume that $\langle \varphi, r \rangle$ was constructed from $\langle \psi, s \rangle$ as in proposition 4.3. Keep $x$ fixed. By the relativization of lemma 2.14 there exist $v$ and $FP^{\alpha, \Sigma^b_1}$-functions $f, g, h$ such that

$$\begin{gathered} f(y) < 1/6, \\ g(y, \bullet)\colon v(1/50 + f(y))s(x) \twoheadrightarrow v \times \{w < s(x) \mid \psi(x, y, w)\}, \\ h(y, \bullet)\colon v(1/50 + 1 - f(y))s(x) \twoheadrightarrow v \times \{w < s(x) \mid \neg\psi(x, y, w)\} \end{gathered}$$

for all $y < q(x)$. By the proof of proposition 4.3 and the relativization of proposition 2.18 there exists a $v'$ and an $FP^{\alpha, \Sigma^b_1}$-function $g'$ such that

$$g'(y, \bullet)\colon v'(r(x)/(4q(x))) \twoheadrightarrow v' \times \{w < r(x) \mid \varphi(x, y, w)\}$$

for all $y < q(x)$. We define $g''(u) = g'(u \bmod q(x), \lfloor \frac{u}{q(x)} \rfloor)$, and observe that

$$g''\colon v'(r(x)/4) \twoheadrightarrow v' \times \{w < r(x) \mid \exists y < q(x)\, \varphi(x, y, w)\},$$

thus

$$\Pr_w(\exists y < q(x)\, \varphi(x, y, w)) \preceq^1_0 1/4. \qquad \square$$

**Proposition 4.5** *(in $T^1_2 + dWPHP(FP^{\Sigma^b_1})$) 1/4-definable prAM-problems are in prNP/poly. I.e., if $L = \langle L^+, L^- \rangle$ is a 1/4-definable prAM-problem, and $n \in \mathrm{Log}$, then there exists a poly-size nondeterministic circuit $C\colon 2^n \to 2$ such that*

$$x \in L^+ \Rightarrow C(x) = 1,$$
$$x \in L^- \Rightarrow C(x) = 0$$

*for every $x < 2^n$.*

*Proof:* Let $\langle \varphi, r \rangle$ be a 1/4-definition of $L$. Using twice the relativized version of lemma 3.10, there exists a circuit $D\colon 2^n \to \{0,1,*\}$ with an $NP$-oracle such that

$$x \in L^+_{\varphi,r,1/4} \to D(x) = 1 \to x \in L^+_{\varphi,r,1/3},$$
$$x \in L^-_{\varphi,r,1/4} \to D(x) = 0 \to x \in L^-_{\varphi,r,1/3}.$$

for every $x < 2^n$. Let $\langle \psi, s \rangle$ be a $2^{-2n}$-definition of $L_{\varphi,r,1/3}$, available by proposition 4.3. For simplicity, we may assume that $s(x) = s$ is constant for all $x < 2^n$. Then

$$\Pr_{w<s}\big((D(x) = 1 \wedge \neg\psi(x,w)) \vee (D(x) = 0 \wedge \psi(x,w))\big) \preceq^1_0 2^{-2n}$$

for every $x < 2^n$. Using the uniformity of the proof of propositions 4.3 and 2.18, we obtain

$$\Pr_w\big(\exists x < 2^n\, (D(x) = 1 \wedge \neg\psi(x,w)) \vee (D(x) = 0 \wedge \psi(x,w))\big) \preceq^1_0 1/2$$

by the same reasoning as in theorem 4.4. By $dWPHP(FP^{\Sigma^b_1})$ there exists $w < s$ such that

$$D(x) = 1 \to \psi(x,w),$$
$$D(x) = 0 \to \neg\psi(x,w)$$

for every $x < 2^n$, and then it suffices to define $C(x) \leftrightarrow \psi(x,w)$. $\square$

As $AM \subseteq BPP^{NP}$, the relativized version of proposition 3.14 implies that every definable $AM$-predicate is in $\Sigma^b_3 \cap \Pi^b_3$. We will formalize the stronger result $AM \subseteq coRP^{NP[1]} \subseteq \Pi^b_2$ from [1]. The proof is based on [20].

**Theorem 4.6** *(in $T^1_2 + dWPHP(FP^{\Sigma^b_1})$) Let $L = \langle L^+, L^- \rangle$ be a 1/4-definable prAM-problem. There exists a $\Sigma^b_1$-formula $\varphi(x,y)$ and a PV-function $r(x)$ such that for every $x$,*

$$x \in L^+ \Rightarrow \forall y \le r(x)\, \varphi(x,y),$$
$$x \in L^- \Rightarrow \Pr_{y \le r(x)}(\varphi(x,y)) \preceq^1_0 1/2.$$

*Proof:* In proposition 4.3, the number of random bits increases polynomially in the number of iterations, but the probability of error decreases exponentially. Thus there exists $\langle \psi, s \rangle$ which is a $1/(4|s(x)|)$-definition of $L$. We may assume $s(x)$ is a power of two. We define

$$r(x) := s(x)^{|s(x)|},$$
$$\varphi(x,y) \leftrightarrow \exists w < s(x)\, \forall i < |s(x)|\, \psi(x, w \oplus y_i),$$

where $y$ is decomposed as a sequence of $|s(x)|$ numbers $y_i < s(x)$, and $\oplus$ is bitwise *XOR*.

Let $x \in L^+$, and fix $y < r(x)$. We have $\Pr_{w<s(x)}(\neg\psi(x,w)) \preceq_0^1 1/(4|s(x)|)$, and $\bullet \oplus y_i$ is a poly-time computable involution on $2^{|s(x)|}$, thus

$$\Pr_w(\neg\psi(x, w \oplus y_i)) \preceq_0^1 \frac{1}{4|s(x)|}$$

for every $i < |s(x)|$. We obtain

$$\Pr_w(\exists i < |s(x)| \, \neg\psi(x, w \oplus y_i)) \preceq_{1/4}^1 \frac{1}{4}$$

from relativization of proposition 2.19, thus $\varphi(x,y)$ by $dWPHP(FP^{\Sigma_1^b})$.

Let $x \in L^-$, and write $s = s(x)$. We have $\Pr_w(\psi(x,w)) \preceq_0^1 1/(4|s|) \leq 1/4$, thus there exists a circuit $C_1$ with an $NP$ oracle such that

$$C_1 \colon v(s/4) \twoheadrightarrow v \times \{u < s \mid \psi(x,u)\}$$

for some $v > 0$. As $w \oplus \bullet$ is a poly-time involution, we have a circuit $C_2$ such that for any $w < s(x)$,

$$C_2(w, \bullet) \colon v(s/4) \twoheadrightarrow v \times \{u < s \mid \psi(x, w \oplus u)\}.$$

We apply $|s|$ copies of $C_2$ in parallel to obtain a circuit $C_3$ such that

$$C_3(w, \bullet) \colon v^{|s|}(s/4)^{|s|} \twoheadrightarrow v^{|s|} \times \{y < s^{|s|} \mid \forall i < |s| \, \psi(x, w \oplus y_i)\},$$

and rearranging the domain yields a circuit $C_4$ such that

$$C_4 \colon v^{|s|}(s/4)^{|s|}s \twoheadrightarrow v^{|s|} \times \{y < s^{|s|} \mid \exists w < s \, \forall i < |s| \, \psi(x, w \oplus y_i)\},$$

thus

$$\Pr_y(\varphi(x,y)) \preceq_0^1 \frac{s}{4^{|s|}} \leq \frac{1}{2}. \qquad \square$$

## References


[1] László Babai, *Trading group theory for randomness*, in: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, 1985, pp. 421–429.

[2] László Babai and Shlomo Moran, *Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes*, Journal of Computer and System Sciences 36 (1988), no. 2, pp. 254–276.

[3] Charles H. Bennett and John Gill, *Relative to a random oracle A,* $\mathbf{P}^A \neq \mathbf{NP}^A \neq$ co-$\mathbf{NP}^A$ *with probability* 1, SIAM Journal on Computing 10 (1981), no. 1, pp. 96–113.

[4] Errett Bishop and Douglas S. Bridges, *Constructive analysis*, Grundlehren der mathematischen Wissenschaften vol. 279, Springer, 1985.

[5] Daniel P. Bovet, Pierluigi Crescenzi, and Riccardo Silvestri, *A uniform approach to define complexity classes*, Theoretical Computer Science 104 (1992), no. 2, pp. 263–283.

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

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

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

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

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

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

[12] John Gill, *Computational complexity of probabilistic Turing machines*, SIAM Journal on Computing 6 (1977), no. 4, pp. 675–695.

[13] Petr Hájek and Pavel Pudlák, *Metamathematics of first-order arithmetic*, Perspectives in Mathematical Logic, Springer, 1993, second edition 1998.

[14] Juris Hartmanis and Lane A. Hemachandra, *Complexity classes without machines: on complete languages for UP*, Theoretical Computer Science 58 (1988), pp. 129–142.

[15] Emil Jeřábek, *Dual weak pigeonhole principle, Boolean complexity, and derandomization*, Annals of Pure and Applied Logic 129 (2004), pp. 1–37.

[16] ———, *The strength of sharply bounded induction*, Mathematical Logic Quarterly 52 (2006), no. 6, pp. 613–624.

[17] Valentine Kabanets, Charles Rackoff, and Stephen A. Cook, *Efficiently approximable real-valued functions*, Technical Report TR00-034, Electronic Colloquium on Computational Complexity, 2000.

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

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

[20] Clemens Lautemann, *BPP and the polynomial hierarchy*, Information Processing Letters 17 (1983), no. 4, pp. 215–217.

[21] Alexis Maciel, Toniann Pitassi, and Alan R. Woods, *A new proof of the weak pigeonhole principle*, Journal of Computer and System Sciences 64 (2002), no. 4, pp. 843–872.

[22] Noam Nisan and Avi Wigderson, *Hardness vs. randomness*, Journal of Computer and System Sciences 49 (1994), no. 2, pp. 149–167.

[23] Jeff B. Paris, Alex J. Wilkie, and Alan R. Woods, *Provability of the pigeonhole principle and the existence of infinitely many primes*, Journal of Symbolic Logic 53 (1988), no. 4, pp. 1235–1244.

[24] Søren M. Riis, *Making infinite structures finite in models of second order bounded arithmetic*, in: Arithmetic, Proof Theory, and Computational Complexity (P. Clote and J. Krajíček, eds.), Oxford Logic Guides vol. 23, Oxford University Press, 1993, pp. 289–319.

[25] Michael Sipser, *A complexity theoretic approach to randomness*, in: Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 1983, pp. 330–335.

[26] Neil Thapen, *The weak pigeonhole principle in models of bounded arithmetic*, Ph.D. thesis, Oxford University, 2002.

[27] ———, *Structures interpretable in models of bounded arithmetic*, Annals of Pure and Applied Logic 136 (2005), no. 3, pp. 247–266.