Random Polynomial Time Computable Functions

Mitsunori Ogiwara*
University of Electro-Communications

Kenneth W. Regan†
SUNY/Buffalo

December 1993

Abstract

Many randomized algorithms M in the literature have the following features: M may produce different valid outputs for different random strings, may output erroneous values, and/or may fail to give any output at all. This paper formalizes and studies these features, and compares the probabilistic function classes thus defined to language classes such as BPP, RP, and ZPP. The two main problems we study are whether the distribution of outputs can be skewed in favor of one valid value, and whether the probability that M behaves correctly can be amplified. We show that if a certain symmetry between two values in fully-polynomial randomized approximation schemes can be broken, then the answer to the former is yes, and we prove many cases in which the answer to the latter is no.

Note: This paper was originally submitted to the Complexity 1994 conference, before the first author adopted "Ogihara" as his Roman spelling. Our e-mails are now `ogihara@cs.miami.edu`, `regan@buffalo.edu`.

*Supported in part by the NSF under grant CCR-9002292 and the JSPS under grant NSF-INT-9116781/JSPS-ENG-207. E-mail: ogiwara@cso.cs.uec.ac.jp.

†Supported in part by the NSF under grant CCR-9011248. *Address for correspondence:*
Dr. Kenneth W. Regan, Computer Science Department, 226 Bell Hall, UB North Campus, Buffalo, NY 14260-2000.
Email: *regan@cs.buffalo.edu*, tel.: (716) 645–3189, fax: (716) 645–3464.

1 Introduction

The purpose of this paper is to investigate systematically the many kinds of "randomized algorithms" which have been developed in the literature. These algorithms may produce different outputs on different random inputs, or may produce no outputs at all. Right away this requires attention to concepts of *multivalued* and *partial* functions, of *types of error* the algorithms may make, and of *bounds* on these errors.

Consider the following two randomized algorithms in finite fields, specialized to the fields $\mathrm{GF}(2^n)$ for $n \geq 1$. The first, given by von zur Gathen and Giesbrecht [vzGG90], searches for an element u in $\mathrm{GF}(2^n)$ which is *normal* over $\mathrm{GF}(2)$, meaning that the elements $\{ u, u^2, u^4, \ldots, u^{2^{n-1}} \}$ form a basis for $\mathrm{GF}(2^n)$ as vector space over $\mathrm{GF}(2)$. The second one searches for an element u which is *primitive*, meaning that the set $\{ u, u^2, u^3, \ldots, u^{2^n-1} \}$ of powers of u runs through all the non-zero elements of $\mathrm{GF}(2^n)$.

M_1: Input 0^n
Flip n coins to form an element $u \in \mathrm{GF}(2^n)$.
Run the *Hensel test* (see [vzGG90]) to determine whether u is normal.
If the test passes, output u, else output $\perp$.

M_2: Input 0^n
Run the deterministic polynomial-time procedure of Shoup [Sho92], which forms a set S_n of polynomial size such that at least one element of S_n is primitive.
Flip coins to select an element u of S_n.
Output u and halt.

M_1 computes the multivalued function $\mathit{normal}(0^n) \mapsto u$ if u is normal in $\mathrm{GF}(2^n)$ over $\mathrm{GF}(2)$. Its success probability on one trial is at least $1/\log n$, according to estimates in [vzGG90]. The Hensel test for $\mathrm{GF}(2^n)$ is computable in polynomial time, and characterizes normal elements. Hence it is possible to *amplify* the success probability to $1 - 2^{-q(n)}$, q a polynomial, by doing $\lceil (\log n) q(n) \log_e 2 \rceil$ repeated trials of M_1. The resulting randomized algorithm M_1' runs in polynomial time, never outputs an erroneous value, and outputs $\perp$ with *exponentially vanishing* (*ev*) error probability.

However, no counterpart of the Hensel test is known for the multivalued function $\mathit{prim}(0^n) \mapsto u$ if u is primitive in $\mathrm{GF}(2^n)$. Instead, $M_2(0^n)$ is prone to make a *mapping error* (ME); that is, output an incorrect value when a correct value exists. Since S has polynomial size, we can at least say that the success probability is non-negligible, or equivalently, that the error is *non-prohibitive* (*np*). But we know of no way to amplify it. Some theoretical issues of practical importance are:

(1) When can probabilities be amplified?

(2) For all x, can a *single* element of *set-f*(x) be found with high probability?

We call (2) the problem of *monic selection*; this is not known to be possible for finding normal elements and many similar functions. This paper undertakes a formal study of these questions. We present a uniform notation system for all the concepts in this introduction, and give some absolute separation results among the classes of partial functions so defined. We tie certain other questions to long-standing open problems about language classes.

2 Preliminaries

In this paper, strings given as inputs, typically denoted by $x, y, z, w, \ldots$ are defined over some alphabet Σ which includes, but need not equal, $\{0,1\}$. Random strings, typically denoted by r, are over $\{0,1\}$ only. The empty string is denoted by λ. If x is an initial substring (i.e., prefix) of y, then we write $x \sqsubseteq y$. The length of a string x is denoted by $|x|$, the cardinality of a set S by $\|S\|$. By "polynomials" we always mean strictly increasing polynomials.

Formally, a partial multivalued function f is the same as a function *set-f* from Σ^* to finite subsets of Σ^*. We write $f(x) \mapsto y$ if $y \in$ *set-f*(x), and say that y is a value of f. We will sometimes regard functions f as taking values in the integers $\mathbf{Z}$ or the non-negative integers $\mathbf{N}$ rather than Σ^*. If for all x, $\|$*set-f*$(x)\| \leq 1$, then f is *(partial) single-valued*. The *domain* of f is defined by $dom(f) = \{ x : \textit{set-}f(x) \neq \emptyset \}$. If $dom(f) = \Sigma^*$, then f is *total*. The *graph* of f is defined by $graph(f) = \{ \langle x, y \rangle : f(x) \mapsto y \}$, where $\langle \cdot, \cdot \rangle$ is some pairing function which is computable and invertible in linear time.

We take for granted in this paper that multivalued functions f are *polynomial length bounded*, meaning that there is a polynomial q such that whenever $y \in$ *set-f*(x), $|y| \leq q(|x|)$. With this in mind, Selman's class NPMV [Sel82] can be defined as $\{ f : graph(f) \in \mathrm{NP} \}$. In naming our probabilistic function classes, we follow the general notational scheme of [Sel91] exemplified by the following:

- NPSV $= \{ f \in \mathrm{NPMV} : f$ is single-valued $\}$.
- $\mathrm{NPMV}_g = \{ f \in \mathrm{NPMV} : graph(f) \in \mathrm{P} \}$.
- $\mathrm{NPMV}_t = \{ f \in \mathrm{NPMV} : f$ is total $\}$.

These modifiers can be combined: NPSV_g, NPMV_{gt}, and so on. PF denotes the class of deterministic polynomial time computable total functions.

If $dom(f) = dom(g)$ and $graph(f) \subseteq graph(g)$, then f is a *refinement* of g. Finally, for any classes $\mathcal{F}$ and $\mathcal{G}$ of multivalued partial functions, we write $\mathcal{G} \subseteq_c \mathcal{F}$ to signify that every function in $\mathcal{G}$ has a refinement in $\mathcal{F}$. For example, the function *sat* which maps a Boolean formula x to y iff y is a satisfying assignment of x belongs to NPMV_g, but not to NPMV_t. It is known that NPMV $\subseteq_c$ NPSV iff *sat* has a refinement in NPSV, and this happens only if the polynomial hierarchy collapses to Σ_2^p [HNOS93].

3 Probabilistic Transducers and Function Classes

The following formalizes the model used widely in the literature, and conveniently lets us write $\Pr_r[M(x,r) = y]$ with r understood as drawn uniformly from $\{0,1\}^{p(|x|)}$.

Definition 3.1. A *probabilistic polynomial-time transducer* is a deterministic Turing machine M which has two input tapes, one for the *input* x and one for the *random string* r, and which runs in time $p(|x|)$, where p is a polynomial. On any input x of length n, r is chosen under uniform distribution from $\{0,1\}^{p(n)}$. $M(x,r)$ may either *accept* and output a *value* $y \in \Sigma^*$, or *reject* and output $\perp$. For short we call M a *p-machine.*

Our intent is that M probabilistically computes a partial function $f(x)$ of one variable. In actuality, M deterministically computes a polynomial-time function $g(x,r)$ of two variables; if we count $\perp$ as a value, then g is total. There are several kinds of errors which a p-machine M can make on a given input x and random r:

Erroneous Rejection: $x \in dom(f)$, but $M(x,r)$ rejects.
Erroneous Acceptance: $x \notin dom(f)$, but $M(x,r)$ accepts.
Mapping Error: $x \in dom(f)$, but $M(x,r)$ returns a string $y \notin set\text{-}f(x)$.

These three kinds of errors are mutually exclusive. We write $\Pr_{\rm ME}(M,f,x)$ for $\Pr_r[M$ on input x makes a mapping error], and $\Pr_{\rm ER}(M,f,x)$, $\Pr_{\rm EA}(M,f,x)$ similarly for erroneous rejection or acceptance. We say that M makes a *domain error* if it makes either a rejection or acceptance error. Having these three kinds of error is intended as the function-class analogue of "1-sided" or "2-sided" error for languages.

Definition 3.2. Let e_r, e_a, and e_m be functions from $\mathbf{N}$ into $[0,1]$ which stand for tolerated probabilities of error. Then a multivalued partial function f belongs to $\mathrm{PRMV}(e_r,e_a,e_m)$ if there is a p-machine M such that for all inputs x, $\Pr_{\rm ER}(M,f,x) \le e_r(|x|)$, $\Pr_{\rm EA}(M,f,x) \le e_a(|x|)$, and $\Pr_{\rm ME}(M,f,x) \le e_m(|x|)$.

If f is partial single valued, we write $f \in \mathrm{PRSV}(e_r,e_a,e_m)$. If for all x, $\|set\text{-}f(x)\| \le 2$, then $f \in \mathrm{PR2V}(e_r,e_a,e_m)$.

Just as it is useful to speak of "bounded error" for languages in BPP without specifying a particular bound, so we generalize the above in terms of conditions on error probabilities.

Definition 3.3. The following *error bound conditions* on error probabilities $P(n)$, where $P(n)$ stands for $\max_{|x|=n}\Pr_{\rm ER}(M,f,x)$, $\max_{|x|=n}\Pr_{\rm EA}(M,f,x)$, or $\max_{|x|=n}\Pr_{\rm ME}(M,f,x)$, can be used in defining $\mathrm{PRMV}(\cdot,\cdot,\cdot)$ function classes (here q stands for a polynomial):

Name	Abbrev.	Defining condition
No condition	$-$	$(\forall^\infty n)\ P(n) < 1$
Non-prohibitive	np	$(\exists q)(\forall^\infty n)\ P(n) < 1 - 1/q(n)$
Unbounded	u	$(\forall^\infty n)\ P(n) < 1/2$
Bounded	b	$(\exists \epsilon > 0)\ (\forall^\infty n)\ P(n) < 1/2 - \epsilon$
Vanishing	v	$(\forall \epsilon > 0)\ (\forall^\infty n)\ P(n) < \epsilon$
Polynomially vanishing (q fixed)	pv_q	$(\forall^\infty n)\ P(n) < 1/q(n)$
Negligible	n	$(\forall q)\ (\forall^\infty n)\ P(n) < 1/q(n)$
Exponentially vanishing (q fixed)	ev_q	$(\forall^\infty n)\ \mathrm{P}(n) < 2^{-q(n)}$
Zero error	z	$(\forall^\infty n)\ \mathrm{P}(n) = 0$.

Given conditions E_r, E_a, and E_m, f belongs to $\mathrm{PRMV}(E_r,E_a,E_m)$ if there exists a p-machine M such that $\Pr_{\rm ER}(M,f,x)$ satisfies E_r, and similarly for E_a and E_m.

When languages L in BPP are said to have "exponentially vanishing error," the meaning is that for every polynomial q there exists a BPP-machine M such that for all x, $\Pr[M(x) \neq L(x)] \le 2^{-q(|x|)}$. Rather than take one more level of abstraction to formalize this, we appeal to this common parlance to make the meaning of e.g. "$\mathrm{PRMV}(ev,ev,ev)$" clear; *viz.* for each q, there exists an M which meets ev_q conditions on all three kinds of error simultaneously. The following most important examples illustrate the notation.

Definition 3.4. BPMV stands for $\mathrm{PRMV}(b,b,b)$. That is, a partial multivalued function f belongs to BPMV if there exists a p-machine M and $\epsilon > 0$ such that for all $x \in \Sigma^*$,

(i) $x \in dom(f) \Longrightarrow \Pr_r[M(x,r) \in set\text{-}f(x)] > 1/2 + \epsilon$.

(ii) $x \notin dom(f) \Longrightarrow \Pr_r[M(x,r) \neq \perp] < 1/2 - \epsilon$.

If f is total, then (ii) can be deleted, and we can write $f \in \mathrm{PRMV}(b,z,b)$. Note that $f \in \mathrm{PRMV}(b,z,b)$ need not imply that f is total; but one can extend f to a total function f' by defining $f'(x) = 0$ for $x \notin dom(f)$, and make the p-machine M output 0 instead of $\perp$. Doing this introduces one subtlety: previously when $x \in dom(f)$ and $M(x,r) = \perp$ this was classed as erroneous rejection, but now it is a mapping error. To define BPSV for partial single-valued f, replace (i) by

$$x \in dom(f) \Longrightarrow \Pr_r[M(x,r) = f(x)] > 1/2 + \epsilon. \tag{1}$$

Definition 3.5. ZPMV stands for $\mathrm{PRMV}(b,z,z)$. Put another way, $f \in \mathrm{ZPMV}$ if there exist $\epsilon > 0$ and a p-machine M such that for all $x \in \Sigma^*$:

(a) For all $r \in \{0,1\}^{p(n)}$, $M(x,r) \in$ *set-*$f(x)$ or $M(x,r) = \perp$.

(b) $x \in dom(f) \Longrightarrow \Pr_r[M(x,r) = \perp] < 1/2 - \epsilon$.

Similarly, ZPSV stands for $\mathrm{PRSV}(b,z,z)$. The choices of names are justified by results in the next section. First we express the observation that bounded-error probabilities can be *amplified* if either (a) f is single-valued, or (b) $graph(f)$ belongs to P (or to BPP, for that matter). When $graph(f) \in P$, erroneous acceptance and mapping errors need never occur.

Proposition 3.1 *(a)* $\mathrm{BPSV} = \mathrm{PRSV}(ev, ev, ev)$.
(b) BPMV_g, ZPMV_g, *and even* $\mathrm{PRMV}(np, np, np)$, *are all the same as* $\mathrm{PRMV}_g(ev, z, z)$.

The proof is by standard means of taking polynomially many repeated trials, using majority vote in (a). As a corollary to (a) of the known result for BPP, functions in BPSV have polynomial-sized circuits. In Section 6 we prove strong senses in which probabilities cannot be amplified at all for general $\mathrm{PRMV}(\cdot,\cdot,\cdot)$ functions.

4 Single-Valued Functions and Language Classes

We start with some respects in which BPSV and ZPSV are natural analogues of BPP and ZPP. The *characteristic function* of a language A, denoted by χ_A, is defined for all x by $\chi_A(x) :=$ 0 if $x \notin A$, 1 if $x \in A$. The *partial characteristic function* ρ_A instead has $\rho_A(x) =$ undefined if $x \notin A$. Besides $graph(f) = \{\langle x, y\rangle : f(x) \mapsto y\}$, we associate to f the sets

$$\begin{array}{lcl} subgraph(f) & := & \{\langle x,w\rangle : (\exists y)\,[f(x) \mapsto y \ \wedge\ w \leq y]\}, \\ prefs(f) & := & \{\langle x,w\rangle : (\exists y)\,[f(x) \mapsto y \ \wedge\ w \sqsubseteq y\}, \\ code(f) & := & \{\langle x,i,b\rangle : (\exists y)\,[f(x) \mapsto y \ \wedge\ \text{the } i\text{th bit of } y \text{ is } b]\}. \end{array}$$

The first few results are simple, and proofs are omitted. Proposition 4.2 reflects the well-known contrast between $graph(f)$ and $subgraph(f)$, which is familiar from the classes C$_=$P and PP. An example showing that the converse to (a) is unlikely to hold, even when $graph(f) \in$ P, is the following NPSV representation of *factoring*. Define $f(m) = (p_1, c_1, i_1, p_2, c_2, i_2, \ldots, p_k, i_k, c_k)$, where $p_1 < p_2 < \ldots < p_k$, $m = p_1^{i_1} p_2^{i_2} \cdots p_k^{i_k}$, and for each j, $1 \leq j \leq k$, c_j is the unique certificate for the primality of p_j given by the method of Fellows and Koblitz [FK92]. However, $f \in$ BPSV would be very surprising.

Proposition 4.1 *(a) For any* $f \in \mathrm{BPSV}$, $dom(f) \in \mathrm{BPP}$.

(b) For any language A, $A \in \text{BPP} \iff \chi_A \in \text{BPSV}$.
(c) For any language A, $A \in \text{BPP} \iff \rho_A \in \text{BPSV}$.
(d) For any language A, $A \in \text{ZPP} \iff \chi_A \in \text{ZPSV}$.

Proposition 4.2 *For any partial single-valued function f,*

(a) $f \in \text{BPSV} \Longrightarrow \mathit{graph}(f) \in \text{BPP}$.
(b) $f \in \text{ZPSV} \Longrightarrow \mathit{graph}(f) \in \text{ZPP}$
(c) $f \in \text{BPSV} \iff \mathit{subgraph}(f) \in \text{BPP} \iff \mathit{prefs}(f) \in \text{BPP} \iff \mathit{code}(f) \in \text{BPP}$.
(d) $f \in \text{ZPSV} \iff \mathit{subgraph}(f) \in \text{ZPP} \iff \mathit{prefs}(f) \in \text{ZPP} \iff \mathit{code}(f) \in \text{ZPP}$.

Matters become more difficult and interesting when we consider connections to the classes RP and coRP. Recall that a language A belongs to RP iff there is a probabilistic polynomial time TM acceptor N such that for all x, $x \in A \Longrightarrow \Pr[N(x) \text{ accepts}] > 1/2$, and $x \notin A \Longrightarrow \Pr[N(x) \text{ accepts}] = 0$. That is, if $N(x)$ accepts then certainly $x \in A$, while rejection by N may err. The error probability can be made exponentially vanishing by polynomially many repeated trials. One technical point is that the outcome $M(x, r) = \perp$ may come from either the case $x \in \mathit{dom}(f)$ or $x \notin \mathit{dom}(f)$—this already shows up in the question $f \in \text{ZPSV} \stackrel{?}{\Longrightarrow} \mathit{dom}(f) \in \text{ZPP}$, in comparison to (a) in Proposition 4.1. Instead:

Proposition 4.3 *(a) For any $f \in \text{ZPSV}$, $\mathit{dom}(f) \in \text{RP}$.*
(b) For any language A, $A \in \text{RP} \iff \rho_A \in \text{ZPSV}$.

Recall that ZPSV is $\text{PRSV}(b, z, z)$. The subgraph and prefix languages instead lead to:

Proposition 4.4 *Let f be a partial single-valued function.*

(a) If $\mathit{subgraph}(f) \in \text{RP}$ or $\mathit{prefs}(f) \in \text{RP}$, then $f \in \text{PRSV}(b, z, b)$.
(b) $\mathit{subgraph}(f) \in \text{RP} \iff$ there is a p-machine M which computes f within bounds (b, z, b) such that whenever $x \in \mathit{dom}(f)$ and $M(x)$ outputs y, $y \leq f(x)$.
(c) $\mathit{prefs}(f) \in \text{RP} \iff$ there is a p-machine M which computes f within bounds (b, z, b) such that whenever $x \in \mathit{dom}(f)$ and $M(x)$ outputs y, $y \sqsubseteq f(x)$.

The converse to (a) appears not to hold: if $f \in \text{PRSV}(b, z, b)$, the most we seem to have is $\mathit{subgraph}(f) \in \text{BPP}$. For $\mathit{code}(f)$ the situation appears to be quite different, and to depend on whether the exact length of $f(x)$ is known in advance.

Proposition 4.5 *Let f be given, and suppose there is a total function $g \in \text{PF}$ such that for all x, if $x \in \mathit{dom}(f)$ then $g(x) = |f(x)|$ in unary. Then $f \in \text{ZPSV} \iff \mathit{code}(f) \in \text{RP} \iff \mathit{code}(f) \in \text{coRP} \iff \mathit{code}(f) \in \text{ZPP}$.*

The subgraph and prefix languages give results symmetrical to Proposition 4.4 (b) and (c) in regard to coRP:

Proposition 4.6 *Let f be a partial single-valued function.*

(a) $\mathit{subgraph}(f) \in \text{coRP} \iff$ there is a p-machine M which computes f within bounds (z, b, b) such that whenever $x \in \mathit{dom}(f)$ and $M(x)$ outputs y, $y \geq f(x)$.
(b) $\mathit{prefs}(f) \in \text{coRP} \iff$ there is a p-machine M which computes f within bounds (z, b, b) such that whenever $x \in \mathit{dom}(f)$ and $M(x)$ outputs y, $f(x) \sqsubseteq y$.

However, when $code(f) \in$ coRP and the hypothesis on g in Proposition 4.5 is absent, we can only deduce that $f \in \mathrm{PRSV}(z, b, b)$. For multivalued functions, we have:

Theorem 4.7 *The following are equivalent:*

(a) $\mathrm{NP} = \mathrm{RP}$,
(b) $\mathrm{NPMV} \subseteq \mathrm{PRMV}(np, -, np)$,
(c) $\mathrm{NPMV} \subseteq \mathrm{PRMV}(n, np, -)$,
(d) $\mathrm{NPMV} \subseteq \mathrm{PRMV}(np, n, -)$,
(e) $\mathrm{NPMV} \subseteq \mathrm{PRMV}(ev, z, z)$.

The proofs of (b) and (c) use the self-reducibility structure of SAT, and are related to results of Adleman and Manders [AM77]. The most interesting immediate question is the relationship between BPMV and NPMV.

Proposition 4.8 *Under any of the above definitions of* BPMV*,*

$$\mathrm{BPMV} \subseteq \mathrm{NPMV} \Longrightarrow \mathrm{BPMV} \subseteq_c \mathrm{NPMV} \Longrightarrow \mathrm{BPSV} \subseteq \mathrm{NPSV} \Longrightarrow \mathrm{BPP} \subseteq \mathrm{NP}.$$

Proposition 4.9 *If* $\mathrm{BPMV} \subseteq_c \mathrm{BPSV}$ *and* $\mathrm{BPP} \subseteq \mathrm{NP}$ *then* $\mathrm{BPMV} \subseteq_c \mathrm{NPSV}$.

5 Monic Selection and Symmetry-Breaking

Suppose we have a BPMV algorithm M for a function f such as finding normal elements in a field. Can we replace M by a p-machine M' with the property that for all $x \in dom(f)$, there is a single $y \in$ *set-*$f(x)$ such that $\Pr_r[M(x, r) = y] > 3/4$? If so, we say that f allows *monic selection*. Formally, this is the same as saying that f has a refinement in BPSV (note that the "3/4" can be amplified). The most common examples of BPMV functions f have $graph(f) \in \mathrm{P}$, and then the question of monic selection becomes: Is $\mathrm{BPMV}_g \subseteq_c \mathrm{BPSV}$?

A "yes" answer would imply that two users running M' on separate machines would with high probability find the same y. Thus our question seems related to important issues in distributed consensus (albeit with no element of communication), but several inquiries have not turned up a prior reference as of this writing. Much previous work (e.g. [JVV86, GMS87]) has been devoted to the problem of selection with uniform distribution on the solution space, but we have not seen comparable studies of how far the distribution can be *biased*. Our requirement is also different from the notion of probabilistically "isolating a unique element" in Chari, Rohatgi, and Srinivasan [CRS93]. They use the method from [MVV87] of assigning random weights to edges so that with high probability, there is a unique minimum-weight perfect matching (when one exists at all), but different random weightings can yield different matchings.

The following natural example illustrates the difficulty of our question even when there are only two possible values. A function $f : \Sigma^* \to \mathbf{N}$ is said to have a *fully polynomial time randomized approximation scheme* (fpras) [KL83, JVV86] if there is a p-machine M such that for all $x \in \Sigma^*$ and $\epsilon > 0$ (where we suppose $\epsilon = 1/c$ for some integer c):

$$\Pr_r \left[\frac{f(x)}{(1+\epsilon)} \;\leq\; M(\langle x, 0^c \rangle, r) \;\leq\; f(x)(1+\epsilon) \right] \;>\; 3/4. \tag{2}$$

Jerrum and Sinclair [JS89] showed that the permanent function for 0-1 matrices, which is #P-complete [Val79], has an fpras. Note that M is multi-valued. We observe that the approximation

can be done by a total function which is at most 2-valued. The "3/4" here and in (2) can be amplified to give exponentially vanishing error.

Proposition 5.1 *Let f have an fpras. Then there is a p-machine M' such that for all $x \in \Sigma^*$ and $c > 0$, there are two values y_1, y_2 such that $f(x)/(1+\epsilon) \leq y_1 \leq y_2 \leq f(x)(1+\epsilon)$ and $\Pr_r[M'(x, 0^c) \in \{ y_1, y_2 \}] > 3/4$.*

The proof idea is to let $u = M(x, r)$ and round u off to the nearest of appropriately-chosen gridpoints. However, if the true value of $f(x)$ is midway between gridpoints, then we may expect "equal scatter" between the two values, with no non-negligible advantage for either. If instead we always "round down," then we have a similar situation when $f(x)$ is close to a gridpoint. We call the problem of whether M' can be made single-valued the "symmetry-breaking problem for fpras."

We first show that if monic selection is possible between two values, then it is possible from among exponentially many. Let PR2V(ev) abbreviate $\mathrm{PR2V}_t(z, z, ev)$.

Theorem 5.2 *If* $\mathrm{PR2V}_t(ev) \subseteq_c \mathrm{BPSV}$, *then* $\mathrm{BPMV}_g \subseteq_c \mathrm{BPSV}$.

The proof is not so simple as that for the analogous result $\mathrm{NP2V} \subseteq_c \mathrm{NPSV} \Longrightarrow \mathrm{NPMV}_g \subseteq_c \mathrm{NPSV}$ (see [Sel91, HNOS93]. One attempt is to let $f \in \mathrm{BPMV}_g$ be given, let M be a p-machine computing f, and by analogy with the next-ID function of an NPMV machine, define $g(x, u) \mapsto b$ if $(\exists r \sqsupseteq ub)\, M(x, r) \in \textit{set-}f(x)$. (Here $b \in \{ 0, 1 \}$.) However, u might be a node in the tree of M with very few valid outputs below it, and so g might not be in BP2V. A second attempt is to define $g(x, u) \mapsto 1$ if $\Pr_{r \sqsupseteq u}[M(x, r) \in \textit{set-}f(x)] \geq 1/4$, and $g(x, u) \mapsto 0$ if $\Pr_{r \sqsupseteq u}[M(x, r) \in \textit{set-}f(x)] \leq 3/4$. Then g is total and does belong to PR2V(ev), so by hypothesis there is a total single-valued restriction g' and an M' which computes it with high probability. However, depth-first backtrack search on '1' values of g' might take exponential time. Our proof modifies the second definition to make the search halt in expected polynomial time.

Proof Sketch. Given f and the p-machine M, let $q(n) = 2p(n) + 5$. For all a, $0 \leq a \leq p(n) + 1$, and all $u \in \{ 0, 1 \}^{<p(n)}$, define

$$g(x, u) \mapsto a \quad \text{if} \quad \Pr_{r \sqsupseteq u}[M(x, r) \in \textit{set-}f(x)] \in [\frac{2a}{q(n)} \ \ldots \ \frac{2a+3}{q(n)}].$$

This covers $[0 \ldots 1]$ with $p(n) + 1$ intervals so that adjacent intervals overlap, but no point is in more than two intervals and there is a large gap between every second interval. Then g is total. Since $\textit{graph}(f) \in \mathrm{P}$, one can estimate $\Pr_{r \sqsupseteq u}[M(x, r) \in \textit{set-}f(x)]$ to within an additive term of $1/p(n)^2$ with high probability by taking polynomially many trials. Hence $g \in \mathrm{PR2V}_t(ev)$. By hypothesis, g has a single-valued restriction in $g' \in \mathrm{BPSV}$. The probability of error in g' can be made exponentially vanishing in polynomial time, so that with high probability, a search which requests polynomially many values of g' never obtains an erroneous one. The conclusion follows from the observation that if $g'(x, u) = a$, then at least one child v of u has $g'(x, v) \geq a - 1$. The root has value $g'(x, \lambda) = p(n) + 1$. Hence the path which takes the left child iff its value is at most one less than the current node hits the bottom before the probability reaches zero. □

The attempt to do the left-leaning path directly with g again runs into symmetry-breaking problems if the value $g(x, v)$ of the left child of u is in the overlap between "one less" and 'two less." Now we observe:

Theorem 5.3 *If the symmetry-breaking problem can be solved for fpras (for #P functions), then every function in* BPMV_g *allows monic selection.*

Proof Sketch. Let $f \in \mathrm{BPMV}_g$ be given, and with reference to the last proof, define

$$h(x,u) = 2^{p(n)} + 2^{|u|} \cdot \|\{\, r \in \{\,0,1\,\}^{p(n)} : r \sqsupseteq u \ \wedge\ M(x,r) \in \mathit{set\text{-}f}(x)\,\}\|.$$

Then $h \in \#\mathrm{P}$. We claim that thanks to the padding term $2^{p(n)}$, h has an fpras computable by sampling polynomially many values as in the previous proof. (Before, g only estimated the number of witnesses below node u additively, not up to a multiplicative factor of $(1+\epsilon)$, and might give zero if the number were small, but now that the numbers are between $2^{p(n)}$ and $2^{p(n)+1}$, the factor pulls off a large interval.) Taking $\epsilon \approx 1/p(n)$ makes it possible to cover $[2^{p(n)} \ldots 2^{p(n)+1}]$ by $p(n)+1$ overlapping intervals of roughly equal size whose endpoints are powers of $(1+\epsilon)$. Symmetry breaking for the fpras allows monic selection of these endpoints, which then plays the role of g' in the previous proof. □

We have not been able to find any interesting "collapses," even of BPP into RP or coRP, from the hypothesis $\mathrm{BPMV}_g \subseteq \mathrm{BPSV}$. In the next section we show unconditionally that $\mathrm{BP2V} \not\subseteq_c \mathrm{BPSV}$, and that probabilities in BPMV (without the condition $\mathit{graph}(f) \in \mathrm{P}$) cannot be amplified in general.

6 Containments and Separations Among General PRMV Classes

Say that a p-machine M *covers* a multivalued function f if for all $x \in \mathit{dom}(f)$, every value of f has nonzero probability, i.e. $(\forall y \in \mathit{set\text{-}f}(x))\, \Pr_r[M(x,r) = y] > 0$. Adhering to this requirement prevents one from playing undue tricks with the range of f in the diagonalization results which follow. First we observe that if the probability of erroneous acceptance or rejection is bounded, it can be made exponentially small via majority-vote.

Proposition 6.1 *Let M be a p-machine, and suppose that there exist nonnegative constants $c, d < \frac{1}{2}$ such that for every x, $\mathrm{Pr}_{\mathrm{EA}}(M,f,x) \leq \frac{1}{2} - c$ and $\mathrm{Pr}_{\mathrm{ER}}(M,f,x) \leq \frac{1}{2} - d$. Then for any polynomial p, there is a p-machine N such that for every x:*

1. $\mathrm{Pr}_{\mathrm{EA}}(N,f,x) \leq 2^{-p(|x|)}$. *In particular,* $\mathrm{Pr}_{\mathrm{EA}}(N,f,x) = 0$ *if* $\mathrm{Pr}_{\mathrm{EA}}(M,f,x) = 0$.
2. $\mathrm{Pr}_{\mathrm{ER}}(N,f,x) \leq 2^{-p(|x|)}$. *In particular,* $\mathrm{Pr}_{\mathrm{ER}}(N,f,x) = 0$ *if* $\mathrm{Pr}_{\mathrm{ER}}(M,f,x) = 0$.
3. $\mathrm{Pr}_{\mathrm{ME}}(N,f,x) = \mathrm{Pr}_{\mathrm{ME}}(M,f,x)$.
4. *If M covers f at x, then so does N.*

Corollary 6.2 *For any error bound condition E, the following relationships hold in a manner that also preserves the property of f being covered by machines meeting the bounds.*

1. $\mathrm{PRMV}(u,-,E) \subseteq \mathrm{PRMV}(ev,-,E)$
2. $\mathrm{PRMV}(-,-,E) \subseteq \mathrm{PRMV}(-,ev,E)$
3. $\mathrm{PRMV}(b,u,E) \subseteq \mathrm{PRMV}(ev,ev,E)$
4. $\mathrm{PRMV}(u,b,E) \subseteq \mathrm{PRMV}(ev,ev,E)$
5. $\mathrm{PRMV}(b,z,E) \subseteq \mathrm{PRMV}(ev,z,E)$
6. $\mathrm{PRMV}(z,b,E) \subseteq \mathrm{PRMV}(z,ev,E)$

Next we say when, given a p-machine for f, one can build a new p-machine that covers f.

Lemma 6.3 *Let $f \in \mathrm{MV}$ and M be a p-machine. Let q be an arbitrary polynomial. Then there is a p-machine N such that for every x, the following conditions are satisfied:*

1. $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) = \mathrm{Pr}_{\mathrm{EA}}(M, f, x)$;
2. $\mathrm{Pr}_{\mathrm{ER}}(N, f, x) = \mathrm{Pr}_{\mathrm{ER}}(M, f, x)$;
3. $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \leq \mathrm{Pr}_{\mathrm{ME}}(M, f, x) + 2^{-q(|x|)}$; *and*
4. *If* $\mathrm{Pr}_r[M(x, r)$ *accepts*$] > 0$, *then N covers f at x.*

Corollary 6.4 *For any error bounds (E_1, E_2, E_3) with $E_3 \neq z$, $\mathrm{PRMV}(E_1, E_2, E_3)$ is unchanged by the requirement that p-machines meeting these bounds must also cover f.*

For the lone exception to Corollary 6.4, we use a new symbol z^* to state that a p-machine M computes f with zero mapping error and also covers f. The bound z^* will not be used for accepting error bounds or rejecting error bounds. Let $\mathcal{E}$ stand for the set of error bounds in Definition 3.3, and let $\mathcal{E}^{(*)} = \mathcal{E} \cup \{z^*\}$. By Corollaries 6.2 and 6.4, every PRMV class coincides with one of either

- $\mathrm{PRMV}(u, u, E)$ with $E \in \mathcal{E}^{(*)}$.
- $\mathrm{PRMV}(E_1, E_2, E_3)$ with $E_1, E_2 \in \{\, z, ev, - \,\}$ (not both $-$) and $E_3 \in \mathcal{E}^{(*)}$.

Now we consider a problem of whether, given a p-machine that computes a function f, one can reduce the mapping error probability of the machine without increasing the probability of a domain error. Recall $\mathrm{Pr}_{\mathrm{DE}}(M, f, x) = \mathrm{Pr}_{\mathrm{ER}}(M, f, x) + \mathrm{Pr}_{\mathrm{EA}}(M, f, z)$.

Lemma 6.5 *Let s and t be recursive functions of natural numbers to $[0, 1)$, p be a polynomial, and $r : \mathbf{N} \mapsto \mathbf{N}$ be a recursive function. Suppose that for all but finitely many n, $t(n) < r(n) \cdot 2^{-p(n)} \leq s(n)$. Then there is a total multivalued function f such that:*

1. *For some p-machine N and all x, $\mathrm{Pr}_{\mathrm{DE}}(N, f, x) = 0$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \leq r(n) \cdot 2^{-p(n)} \leq s(n)$.*
2. *For all N and x, either $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) = 1$ or $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) > t(|x|)$.*

Theorem 6.6
1. $\mathrm{PRMV}(z, z, -) \not\subseteq \mathrm{PRMV}(-, -, u)$.
2. $\mathrm{PRMV}(z, z, u) \not\subseteq \mathrm{PRMV}(-, -, b)$.
3. $\mathrm{PRMV}(z, z, b) \not\subseteq \mathrm{PRMV}(-, -, v)$.
4. $\mathrm{PRMV}(z, z, v) \not\subseteq \mathrm{PRMV}(-, -, pv)$.
5. $\mathrm{PRMV}(z, z, pv) \not\subseteq \mathrm{PRMV}(-, -, n)$.
6. $\mathrm{PRMV}(z, z, n) \not\subseteq \mathrm{PRMV}(-, -, ev)$.
7. $\mathrm{PRMV}(z, z, z) \not\subseteq \mathrm{PRMV}(-, -, z^*)$.

A similar diagonalization idea, constructing a 3-valued g whose values are "close to equally-likely," so that all 2-valued refinements of g belong to BP2V, produces the following:

Theorem 6.7 $\mathrm{PR2V}(b, b, b) \not\subseteq_c \mathrm{BPSV}$.

The one separation missing from Theorem 6.6 is whether $\mathrm{PRMV}[(z, z, ev)] \not\subseteq \mathrm{PRMV}[(-, -, z)]$, and this remains open. We show that separating PRMV classes with exponentially vanishing mapping error from those with zero mapping error probability ties in to other unsolved problems in about complexity classes, and investigate the relationships between PRMV classes having the same mapping error bounds.

Theorem 6.8 *If* $\mathrm{PP} = \mathrm{RP}$, *then* $\mathrm{PRMV}(-, -, ev) \subseteq \mathrm{PRMV}(ev, -, z)$.

Theorem 6.9 *Let $E \in \mathcal{E}$ be an arbitrary error bound condition.*

1. *If* $\mathrm{PRMV}(u, u, E) \subseteq \mathrm{PRMV}(ev, ev, E)$, *then* $\mathrm{PP} = \mathrm{BPP}$.
2. *If* $\mathrm{PRMV}(ev, ev, E) \subseteq \mathrm{PRMV}(ev, z, E) \cup \mathrm{PRMV}(z, ev, E)$, *then* $\mathrm{BPP} = \mathrm{RP}$.
3. *If either* $\mathrm{PRMV}(ev, z, E) \subseteq \mathrm{PRMV}(z, ev, E)$ *or* $\mathrm{PRMV}(z, ev, E) \subseteq \mathrm{PRMV}(ev, z, E)$, *then* $\mathrm{RP} = \mathrm{coRP}$.
4. *If* $\mathrm{PRMV}(z, ev, E) \cap \mathrm{PRMV}(ev, z, E) \subseteq \mathrm{PRMV}(z, z, E)$, *then* $\mathrm{RP} \cap \mathrm{coRP} = \mathrm{P}$.
5. *If either* $\mathrm{PRMV}(z, ev, E) \subseteq \mathrm{PRMV}(z, z, E)$ *or* $\mathrm{PRMV}(ev, z, E) \subseteq \mathrm{PRMV}(z, z, E)$, *then* $\mathrm{RP} = \mathrm{coRP} = \mathrm{P}$.

Theorem 6.10 1. *If* $\mathrm{PP} = \mathrm{BPP}$, *then for every* $E \in \mathcal{E}$, $\mathrm{PRMV}(u, u, E) \subseteq \mathrm{PRMV}(ev, ev, E)$.
2. *If* $\mathrm{BPP} = \mathrm{RP}$, *then* $\forall E \in \mathcal{E}^{(*)}$: $\mathrm{PRMV}(ev, ev, E) = \mathrm{PRMV}(z, ev, E) \cup \mathrm{PRMV}(ev, z, E)$.
3. *If* $\mathrm{RP} \cap \mathrm{coRP} = \mathrm{P}$, *then* $\forall E \in \mathcal{E}^{(*)}$: $\mathrm{PRMV}(ev, z, E) \cap \mathrm{PRMV}(z, ev, E) = \mathrm{PRMV}(z, z, E)$.
4. *If* $\mathrm{RP} = \mathrm{P}$, *then* $\forall E \in \mathcal{E}^{(*)}$: $\mathrm{PRMV}(ev, z, E) = \mathrm{PRMV}(z, ev, E) = \mathrm{PRMV}(z, z, E)$.

7 Conclusion

We offer our concepts and formalism as useful tools for further study. One open problem is to draw further consequences from the hypothesis that monic selection is possible for all BPMV_g functions. In particular, if a relativized form holds for all functions in $\mathrm{BPMV}_g^{\mathrm{NP}}$, does the polynomial hierarchy collapse? The idea is to combine the familiar Valiant-Vazirani reduction with [HNOS93]. Another question is how the results in Section 6 relate to theorems of Rohatgi et al. [CKR91, CR93, Roh93] which show that tiny amplifications in randomized reductions *to certain languages* would collapse the hierarchy.

References

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractability. In *Proc. 9th Annual ACM Symposium on the Theory of Computing*, pages 151–163, 1977.

[CKR91] R. Chang, J. Kadin, and P. Rohatgi. Connections between the complexity of unique satisfiability and the threshold behavior of randomized reductions. In *Proc. 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 255–269, 1991. J. Comp. Sys. Sci., to appear.

[CR93] S. Chari and P. Rohatgi. On completeness under random reductions. In *Proc. 8th Annual IEEE Conference on Structure in Complexity Theory*, pages 176–184, 1993.

[CRS93] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation, with applications to perfect matching and related problems. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 458–467, 1993.

[FK92] M. Fellows and N. Koblitz. Self-witnessing polynomial-time complexity and prime factorization. In *Proc. 7th Annual IEEE Conference on Structure in Complexity Theory*, pages 107–109, 1992.

[GMS87] O. Goldreich, Y. Mansour, and M. Sipser. Interactive proof systems: provers that never fail and random selection. In *Proc. 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 449–461, 1987.

[HNOS93] L. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. Computing solutions uniquely collapses the polynomial hierarchy. Technical Report CS-TR 93-28, Computer Science Dept., SUNY at Buffalo, August 1993.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18:1149–1178, 1989.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comp. Sci.*, 43:169–188, 1986.

[KL83] R. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proc. 24th Annual IEEE Symposium on Foundations of Computer Science*, pages 56–64, 1983.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[Roh93] P. Rohatgi. *On properties of random reductions.* PhD thesis, Cornell University, 1993.

[Sel82] A. Selman. Reductions on NP and p-selective sets. *Theor. Comp. Sci.*, 19:287–304, 1982.

[Sel91] A. Selman. A taxonomy of complexity classes of functions. Technical Report 91–12, Dept. of Comp. Sci., SUNY / Buffalo, 1991. Revised June 1992, to appear in *J. Comp. Sys. Sci.*

[Sho92] V. Shoup. Searching for primitive roots in finite fields. *Mathematics of Computation*, 58:369–380, 1992.

[Val79] L. Valiant. The complexity of computing the permanent. *Theor. Comp. Sci.*, 8:189–201, 1979.

[vzGG90] J. von zur Gathen and M. Giesbrecht. Constructing normal bases in finite fields. *J. Symb. Comput.*, 10:547–570, 1990.

8 Appendix with Selected Proofs

Proof of Proposition 4.4. (a) By hypothesis, there is a polynomial-time probabilistic TM M such that for all $x, w \in \Sigma^*$,

$$\begin{aligned} x \in \mathit{dom}(f) \ \wedge \ w \leq f(x) \ &\Longrightarrow \ \mathrm{Prob}_r[M(\langle x, w\rangle) \text{ accepts}] > 2/3 \\ x \notin \mathit{dom}(f) \ \vee \ w > f(x) \ &\Longrightarrow \ \mathrm{Prob}_r[M(\langle x, w\rangle) \text{ accepts}] = 0. \end{aligned}$$

The p-machine M' on input x first simulates $M(\langle x, \lambda\rangle)$. If M rejects, then M' rejects also. Otherwise M' attempts to find $f(x)$ by binary search. The bounding polynomial p of M' can be set high enough so that by repeating calls to M, M' succeeds with probability at least $2/3$. When $x \in \mathit{dom}(f)$, there is a nonzero chance that M' rejects at the outset. There is also a nonzero chance of mapping error should M incorrectly reject some $\langle x, w\rangle$, causing the binary search to go lower when it should go higher. Note that *any* erroneous value y returned by M' will have $y < f(x)$. When $x \notin \mathit{dom}(f)$, M' always rejects. Hence $f \in \mathrm{PRSV}(b, z, b)$.

(b) The forward part is as in (a). For the converse, suppose all mapping errors made by M are underestimates, and M never accepts when $x \notin \mathit{dom}(f)$. Then define an acceptor M' which on input $\langle x, w\rangle$ simulates $M(x)$. If $M(x)$ rejects, then M' rejects, while if $M(x)$ returns a value y, M' accepts iff $w \leq y$. Then all "accept" outcomes of M' are correct, and the error in "reject" outcomes is bounded, so M' is an RP-machine for $\mathit{subgraph}(f)$. The proof of (c) is similar. □

Proof of Proposition 4.5. Given a p-machine M which computes f within error bounds (b, z, z), let the acceptor M' on input $\langle x, j, b\rangle$ first verify that $j \leq g(x)$ (if not, it rejects), and then run $M(x)$ repeatedly. If and when $M(x)$ accepts and returns a value y, it follows from the bounds (b, z, z) that $y = f(x)$, so M' can halt and render an error-free verdict on whether $\langle x, j, b\rangle \in \mathit{code}(f)$. So $\mathit{code}(f) \in \mathrm{ZPP}$. Conversely, if $\mathit{code}(f) \in \mathrm{RP}$, let the p-machine M on input x first calculate $k := g(x)$, and then repeatedly simulate the RP-machine for $\mathit{code}(f)$ on the arguments

$$\begin{aligned} &\langle x, 1, 0\rangle\langle x, 2, 0\rangle \ldots \langle x, k, 0\rangle \\ &\langle x, 1, 1\rangle\langle x, 2, 1\rangle \ldots \langle x, k, 1\rangle \end{aligned}$$

If and when it receives exactly one 'yes' answer in each column, M halts and outputs the indicated value; if not, M rejects. The polynomial bound p on the runtime of M can be set high enough so that when $x \in \mathit{dom}(f)$, the chance that M gets these yes answers is at least $2/3$. Since 'yes' answers from an RP-machine are always correct, M never makes a mapping error. When $x \notin \mathit{dom}(f)$, the responses are always 'no,' so M correctly rejects. Hence M runs within error bounds (b, z, z).

Similarly, if $\mathit{code}(f) \in \mathrm{coRP}$, M waits until it receives exactly one 'no' answer in each column. □

(*Remark:* Without the ability to compute k, M might allocate a column $k+1$ and then not be able to trust the two 'no' answers it receives in that column. The best conclusion from $\mathit{code}(f) \in \mathrm{RP}$ that we know without this is that $f \in \mathrm{PRSV}(b, z, b)$.

Proof of Proposition 5.1. Let M be the machine for f from the definition of an fpras. Let $s > 3$, $t = 2s+1$, and $c = 1/\epsilon$ be integers. For each x, let $a(x) = f(x)/(1+1/tc)$, $b(x) = f(x) \cdot (1+1/tc)$,

and $m(x)$ be the smallest integer m such that $|\log x - m\log(1+1/sc)| \leq 1/2$. Define M' to be the machine that, on input $\langle x, 0^c\rangle$, simulates M on $\langle x, 0^{tc}\rangle$ to get a value u, computes $v = m(u)$, and outputs $\lfloor (1+1/sc)^v \rfloor$. For any x, with probability $> 3/4$, M' gets a value u in range $[a(x), b(x)]$. Since $b(x)/a(x) = (1+1/tc)^2 > 1+1/sc$, for each x, there is an integer $d(x)$ such that the value v that M' computes is either $d(x)$ or $d(x)+1$ with probability more than $3/4$. So, for each x, there exist integers $y_1(x)$ and $y_2(x)$ with $y_1(x) \leq y_2(x)$ such that with probability $3 > 4$, M' outputs either $y_1(x)$ or $y_2(x)$. Since $(1+1/tc)(1+1/sc) < 1/c$, for any x, $y_2(x) \leq f(x)\cdot(1+1/c)$. By choosing s sufficiently large, for any x, $y_1(x) \geq f(x)/(1+1/c)$. This proves the proposition. □

Proof of Lemma 6.3. Let f, M, and q be as in the hypothesis. Let $Pacc_M(x)$ stand for the probability that the p-machine M on input x accepts and outputs a value. Let p be a polynomial bounding the length of f; that is, for every x and y, if $f(x) \mapsto y$, then $|y| \leq p(|x|)$. Define N to be the machine that, on input x, behaves as follows:

(a) N simulates M on x. If M rejects x, then so does N. Otherwise, N proceeds to (b).

(b) Let y be the output of M that N obtained in (a). N chooses $i, 0 \leq i \leq 2^{q(|x|)}-1$, with equal probability. If $i \neq 0$, then N outputs y and halts. Otherwise, it proceeds to (c).

(c) N chooses $j, 0 \leq j \leq 2^{p(|x|)+1}-1$, with equal probability. N outputs the j-th smallest string in Σ^* and halts.

It is not hard to see that N accepts x with probability 1 after entering step (b). So, for every x, $Pacc_N(x) = \mathrm{Prob}[N \text{ on } x \text{ enters step (b) }] = Pacc_M(x)$. Thus, for every x, (1) and (2) are both satisfied.

Let $x \in dom(f)$. First suppose that $Pacc_M(x) = 0$. Then, N will never accept x, so $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) = 0$, and thus, (3) and (4) are both satisfied. So, suppose that $Pacc_M(x) > 0$. For every w of length $\leq p(|x|)$, N on x outputs w for some random seed. Since f is $p(n)$ length-bounded, this implies that N covers f at x. So, (4) is satisfied. Moreover, the probability that N on x outputs a value not in $f(x)$ is bounded by:

$$\begin{aligned} & Pacc_M(x)\cdot[(1-2^{-q(|x|)})\mathrm{Pr}_{\mathrm{ME}}(M,f,x) + 2^{-q(|x|)}] \\ \leq\ & Pacc_M(x)\cdot[\mathrm{Pr}_{\mathrm{ME}}(M,f,x) + 2^{-q(|x|)}]. \end{aligned}$$

So, $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \leq \mathrm{Pr}_{\mathrm{ME}}(M, f, x) + 2^{-q(|x|)}$. Thus, (3) is satisfied. Hence, for every x, (3) and (4) are both satisfied. This proves the lemma. □

Proof of Lemma 6.5. Let s, t, p, and r be as in the hypothesis. Let m be such that for every $n \geq m$, $t(n) < r(n)\cdot 2^{p(n)} \leq s(n)$.

Below we will construct a total multivalued function f. For every x, $f(x)$ will map only to $i, 1 \leq i \leq 2^{p(|x|)}$, and have at least $2^{p(|x|)} - r(|x|)$ values. Let $M_1, M_2, \cdots$ be an enumeration of all p-machines. The values of f at x are determined as follows:

(Case 1) $|x| < m$:

We define the values of $f(x)$ to be $i, 1 \leq i \leq 2^{p(|x|)}$.

(Case 2) $|x| \geq m$:

Let x be the j-th smallest string of length n.

(Case 2a) $Prej_{M_j}(x) < 1$ and $\mathrm{Prob}[M_j(x) \mapsto i, 1 \le i \le 2^{p(n)}]/Pacc_{M_j}(x) > 1 - t(|x|)$:

Let $l_1, \cdots, l_k (k = 2^{p(n)})$ be an enumeration of $1, \cdots, 2^{p(n)}$ such that for every $h, 1 \le h \le k-1$ $\mathrm{Prob}[M_j(x) \mapsto l_h] \ge \mathrm{Prob}[M_j(x) \mapsto l_{h+1}]$. We define *set-f*$(x) = \{ l_h : h \ge r(n) + 1 \}$.

(Case 2b) $Prej_{M_j}(x) = 1$ or ($Prej_{M_j}(x) < 1$ and $\mathrm{Prob}[M_j(x) \mapsto i, 1 \le i \le 2^{p(n)}]/Pacc_{M_j}(x) \le 1 - t(|x|)$):

We define *set-f*$(x) = \{ i : 1 \le i \le 2^{p(|x|)} \}$.

It is not hard to see that for every x, $f(x)$ has at least $2^{p(|x|)} - r(|x|)$ values and at most $2^{p(|x|)}$ values. Let N be a machine that, for an input x, outputs each $i, 1 \le i \le 2^{p(|x|)}$ with probability $2^{-p(|x|)}$. Then, $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) = 0$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) = r(|x|) \cdot 2^{-p(|x|)} < s(|x|)$. So, the condition (1) is satisfied.

We prove, by way of contradiction, that the condition (2) is satisfied. Assume, to the contrary, that there is a p-machine N such that for all x, $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) < 1$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \le t(|x|)$. Let j be an index such that $N = M_j$, let $n \in \mathbf{N}$ be such that $n \ge m$ and $2^m \ge j$, and x be the j-th smallest string of length n. By our assumption, $\mathrm{Prob}[M_j(x) \mapsto i, i$ is a value of $f(x)]/Pacc_{M_j}(x) > 1 - t(n)$. Since $f(x) \mapsto i$ only if $i \in \{1, \cdots, 2^{p(n)}\}$, we have $\mathrm{Prob}[M_j(x) \mapsto i, 1 \le i \le 2^{p(n)}]/Pacc_{M_j}(x) > 1 - t(n)$. So, Case 2a holds for x when we attempt to diagonalize against M_j. By our choice of the values of $f(x)$, it holds that $\mathrm{Prob}[M_j(x) \mapsto i, i$ is a value of $f(x)]/Pacc_{M_j}(x) \le 1 - r(n) \cdot 2^{-p(n)} < 1 - t(n)$. This is a contradiction. Hence, the condition (2) is satisfied. This proves the lemma. □

Proof of Theorem 6.6. (Parts 3. 4. and 7. only)

3. Let $s(n) = 1/4$ and $t(n) = 1/5$. Obviously, there is a polynomial p and a recursive function $r : \mathbf{N} \to \mathbf{N}$ such that for all n, $t(n) < r(n) \cdot 2^{-p(n)} \le s(n)$. Take such a pair of p and r, and apply Lemma 6.5 to get the function f. By condition (1), there is a p-machine N such that for all x, $\mathrm{Pr}_{\mathrm{DE}}(N, f, x) = 0$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \le 1/4$. So, $f \in \mathrm{PRMV}(z, z, b)$. Moreover, by condition (2), there is no p-machine N such that for all x, $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) < 1$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \le 1/5$. Thus, f cannot be in $\mathrm{PRMV}(-, -, v)$.

4. Let $s(n) = 1/\log n$ and $t(n) = 1/2 \log n$. Obviously, there is a polynomial p and a recursive function $r : \mathbf{N} \to \mathbf{N}$ such that for all n, $t(n) < r(n) \cdot 2^{-p(n)} \le s(n)$. Take such a pair of p and r, and apply Lemma 6.5 to get f. By condition (1), there is a p-machine N such that for all x, $\mathrm{Pr}_{\mathrm{DE}}(N, f, x) = 0$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \le 1/\log n$. Since $s(n)$ is smaller than any positive constant for all but finitely many n, $f \in \mathrm{PRMV}(z, z, v)$. Moreover, by condition (2), there is no p-machine N such that for all x, $\mathrm{Pr}_{\mathrm{EA}}(N, f, x) < 1$ and $\mathrm{Pr}_{\mathrm{ME}}(N, f, x) \le t(|x|) = 1/2 \log |x|$. Since $q(n)/2 \log n$ tends to ∞ as n tends to ∞ for any polynomial $q(n)$, f cannot be in $\mathrm{PRMV}(-, -, pv)$.

7. Let A be a set not in NP. Define a total multivalued function f as follows:

   - $f(x) \mapsto 0$ for all x;
   - $f(x) \mapsto 1$ if and only if $x \in A$; and
   - for every x, $f(x)$ maps only to 0 and 1.

   Define M to be the p-machine that, on input x, outputs 1 with probability 1. For every x, $\mathrm{Pr}_{\mathrm{EA}}(M, f, x) = \mathrm{Pr}_{\mathrm{ER}}(M, f, x) = \mathrm{Pr}_{\mathrm{ME}}(M, f, x) = 0$. So, $f \in \mathrm{PRMV}(z, z, z)$. The function

f cannot be in PRMV$(-,-,z^*)$. Otherwise, there is a p-machine N such that for every x, $N(x) \mapsto 1$ if and only if $x \in A$, which contradicts to our assumption that $A \notin$ NP. This proves the theorem. □

Proof of Theorem 6.8. Assume PP = RP. Then it holds that $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{PP}^{\mathrm{BPP}} \subseteq \mathrm{PP}^{\mathrm{PH}} \subseteq \mathrm{BPP}^{\mathrm{PP}} \subseteq \mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BPP} \subseteq \mathrm{PP} = \mathrm{RP}$.

Let f be a $p(n)$ length-bounded multivalued function in PRMV$[(-,-,ev)]$. There is a p-machine M such that for every $x \in dom(f)$, $Pacc_M(x) > 1$ and $\mathrm{Pr}_{\mathrm{ME}}(M,f,x) < 2^{-p(n)-1}$. Since there are $2^{p(n)+1}-1$ strings of length at most n and $(2^{p(n)+1}-1)^{-1} > 2^{-p(n)-1}$, for every $x \in dom(f)$, there is a string $y, |y| \leq p(|x|)$, such that $\mathrm{Prob}[M(x) \text{ outputs } y]/Pacc_M x > 2^{-p(|x|)-1}$. For each x, let y_x be the smallest $y, |y| \leq p(|x|)$, such that $\mathrm{Prob}[M(x) \text{ outputs } y]/Pacc_M x > 2^{-p(|x|)-1}$. For every $x \in dom(f)$, y_x is the value of $f(x)$; otherwise, $\mathrm{Pr}_{\mathrm{ME}}(M,f,x) > 2^{-p(n)-1}$, which contradicts to our assumption. Define $L = \{(x,w) : y_x < w\}$. Then, $L \in \mathrm{NP}^{\mathrm{PP}}$, so, by our assumption, $L \in$ BPP. Define sets A, B, and C as follows:

$$\begin{aligned} A &= \{(x,i) : |y_x| \geq i\}; \\ B &= \{(x,i) : |y_x| = i\}; \quad \text{and} \\ C &= \{(x,i,b) : b \in \{0,1\}, y_x(i) = b\}, \end{aligned}$$

where $y_x(i)$ denotes the i-th symbol in y_x Since $L \in$ BPP, $A, B, C \in$ RP. Since $A \in$ BPP, there is a probabilistic polynomial-time procedure R_1 that, on input x,

- outputs y_x with high probability if y_x is defined; and
- outputs *undefined* with probability 1 if y_x is undefined.

Since B is in RP, and $|y_x| \leq p(|x|)$, there is a probabilistic polynomial-time procedure R_2 that, given x and i,

- outputs YES with high probability if $|y_x| = i$; and
- outputs NO with probability 1 if $|y_x| \neq i$.

So, by combining R_1 and R_2, we can construct a probabilistic polynomial-time procedure S that, on input x,

- outputs $|y_x|$ with high probability if y_x is defined;
- never outputs m such that $m \neq |y_x|$ if y_x is defined; and
- outputs *undefined* with probability 1 if y_x is undefined.

Now suppose that we have obtained from S, a purported value of $|y_x|$. Call the value m. We attempt to compute y_x using C. By our assumption, $C \in$ RP or $\overline{C} \in$ RP. If $m = |y_x|$, then for every $i, 1 \leq i \leq m$, and for every $b \in \{0,1\}$, it holds that

(*) $y_x(i) = b$ if and only if $(x,i,b) \in C$ if and only if $(x,i,b') \in \overline{C}$, where $b' = 0$ if $b = 1$ and 0 otherwise.

So, there is a probabilistic polynomial-time procedure T that, given x, m, and i with $1 \leq i \leq m = |y_x|$;

- outputs $y_x(i)$ with high probability; and
- never outputs a value not equal to $y_x(i)$.

Then, by using T as a subroutine, after obtaining m, one can compute y_x with high probability if $m = |y_x|$. Therefore, a single-valued function $g = \lambda x.[y_x]$ is in PRMV$[(ev,ev,z)]$. So, $f \in$ PRMV$[(ev,-,z)]$. This proves the theorem. □