Distribution-Valued Functions and Quantum Computation

Stephen A. Fenner*
Yong Zhang*

Computer Science and Engineering Department
University of South Carolina

June 21, 2003

Abstract

Complexity classes related to quantum computing may all be derived from the class FQP, defined by Aharonov, Kitaev, and Nisan [1]. A function $f \in \text{FQP}$ maps an input $x$ to a probability distribution of possible outputs of a given quantum circuit with input state $|x\rangle$. FQP functions can be simulated by functions which count numbers of solutions to NP problems ("counting functions") via matrix multiplication. This fact has led to the tightest known relationships of quantum complexity classes such as BQP and NQP to traditional complexity classes, namely, NQP $=$ coC$_m$P [5, 16] and BQP $\subseteq$ AWPP [8]. We aim to find a containment tighter than the latter by finding a reasonable counting class lying strictly between BQP and AWPP. We define new classes of distribution-valued functions, based on counting functions and matrix multiplication, restricting the matrices in various natural ways (e.g., unitary or stochastic).

*Columbia, SC 29208 USA. E-mail: {fenner|zhang29}@cse.sc.edu. Partially supported by the U.S. Army Research Office under DAAD 190210048.
1 Preliminaries

We assume knowledge of basic complexity theory as found in [13, 12] for example, including the definitions of P, NP, BPP, languages, functions, and machines. We denote the natural numbers (including zero), the integers, the rationals, the reals, and the complex numbers by \( \mathbb{N} \), \( \mathbb{Z} \), \( \mathbb{Q} \), \( \mathbb{R} \), and \( \mathbb{C} \), respectively. We let \( \Sigma = \{0, 1\} \) and let \( \Sigma^* \) be the set of all finite strings over \( \Sigma \). All inputs to functions and to predicates will be elements of \( \Sigma^* \), we use the word “language” to mean any subset of \( \Sigma^* \). We often identify \( \Sigma^* \) with \( \mathbb{N} \) via the standard dyadic representation. For any \( n \in \mathbb{N} \) we let \( \Sigma^n \) denote the set of all strings over \( \Sigma \) of length \( n \), and we sometimes also identify \( \Sigma^n \) with the set \( \{0, \ldots, 2^n - 1\} \subseteq \mathbb{N} \). It should be clear which identification is used when. We denote the empty string by \( \epsilon \), and the concatenation of strings \( x \) and \( y \) by \( xy \). We write \( x \leq y \) to mean \( z \) is a prefix of \( y \). We denote the length of a string \( x \) by \( |x| \).

We also denote the Euclidean norm of a (column) vector \( v \) or complex number \( z \) by \(|v| \) or \(|z| \). There should be no confusion with string length given the context. If \( M \) is a \( m \times n \) matrix, we write \( M_{i,j} \) for the \((i,j)\)th entry of \( M \), for \( 0 \leq i < m \) and \( 0 \leq j < n \), and we write \( M_{j} \) for the \( j \)th column of \( M \). We will use the standard operator norm on square matrices:

\[
\|A\| = \sup_{|v|=1} |Av|.
\]

For any finite set \( X \) we let \( \|X\| \) denote the cardinality of \( X \).

We fix a standard bijection \( \langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \) which is easy to compute and to invert, such as

\[
\langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + y.
\]

This also acts as a bijection \( \Sigma^* \times \Sigma^* \to \Sigma^* \). Although we may write predicates and functions as taking several inputs, using the pairing function we may assume all functions and predicates are monadic.

We let FP be the set of polynomial-time computable functions.

1.1 Complexity Classes

**Definition 1.1** (Valiant [14]) A function \( f : \Sigma^* \to \mathbb{N} \) is in the class \( \#P \) if and only if there is a polynomial \( p \) and a predicate \( R \in \text{P} \) such that, for all
\[ n \in \mathbb{N} \text{ and all } x \in \Sigma^n, \]
\[ f(x) = \|\{ y \in \Sigma^n \mid R(x, y)\}\|. \]

\#P captures the notion of counting the number of solutions to an NP problem. We will use a closely related class GapP \([3]\) which is closed under subtraction, and thus has nicer algebraic properties.

**Definition 1.2 ([3])** A function \( f : \Sigma^* \rightarrow \mathbb{Z} \) is in the class GapP if and only if there are functions \( g, h \in \#P \) such that, for all \( x \in \Sigma^* \),
\[ f(x) = g(x) - h(x). \]

GapP has several equivalent definitions. It also has several important closure properties, which we now describe. We assume a standard identification of \( \Sigma^* \) with \( \mathbb{Z} \).

**Proposition 1.3 ([3])** GapP satisfies the following closure properties:

1. \( FP \subseteq \#P \subseteq \text{GapP} \), and furthermore, if \( f \in FP \) and \( g \in \text{GapP} \), then \( g \circ f \in \text{GapP} \).

2. If \( f \in \text{GapP} \) then \(-f \in \text{GapP} \).

3. If \( g, h \in \text{GapP} \), then \( g + h \in \text{GapP} \). More generally, if \( f \in \text{GapP} \), and \( p \) is a polynomial, then \( s \in \text{GapP} \), where
\[ s(x) = \sum_{y : |y| \leq p(|x|)} f(x, y). \]

4. If \( g, h \in \text{GapP} \), then \( gh \in \text{GapP} \) (\( gh \) is the pointwise product of \( g \) and \( h \), not the composition). More generally, if \( f \in \text{GapP} \) and \( q \) is a polynomial, then \( p \in \text{GapP} \), where
\[ p(x) = \prod_{i=0}^{q(|x|)} f(x, 0^i). \]
5. If \( g \in \text{GapP} \), then the function \( f \) such that

\[
f(x, y) = \left( \frac{g(x)}{|y|} \right)
\]

is in GapP, where

\[
\binom{r}{k} = \frac{r(r-1)(r-2) \cdots (r-k+1)}{k!}
\]

for all \( r \in \mathbb{C} \) and \( k \in \mathbb{N} \).

GapP can provide simple characterizations of many counting complexity classes (see [3] for details), including

- [9] A language \( L \) is in PP if there is an \( f \in \text{GapP} \) such that, for all \( x \in \Sigma^* \),
  
  \( x \in L \iff f(x) > 0. \)

- [15] A language \( L \) is in \( C_\text{P} \) if there is an \( f \in \text{GapP} \) such that, for all \( x \in \Sigma^* \),
  
  \( x \in L \iff f(x) = 0. \)

- [3] A language \( L \) is in SPP if there is an \( f \in \text{GapP} \) such that, for all \( x \in \Sigma^* \),
  
  \( x \in L \implies f(x) = 1 \)
  
  \( x \notin L \implies f(x) = 0. \)

- [3] A language \( L \) is in WPP if there is an \( f \in \text{GapP} \) and a \( g \in \text{FP} \) such that, for all \( x \in \Sigma^* \), \( g(x) \neq 0 \) and
  
  \( x \in L \implies f(x) = g(x) \)
  
  \( x \notin L \implies f(x) = 0. \)

The class AWPP is the most important one for our purposes. This is not the original definition but is equivalent to it.
Definition 1.4 ([10, 4, 3, 7]) A language $L$ is in AWPP if there is an $f \in \text{GapP}$ and a polynomial $p$ such that, for all $n \in \mathbb{N}$ and $x \in \Sigma^n$, $p(n) \geq 0$ and

$$x \in L \implies \frac{2}{3} \leq \frac{f(x)}{2^p(n)} \leq 1$$

$$x \not\in L \implies 0 \leq \frac{f(x)}{2^p(n)} \leq \frac{1}{3}.$$

AWPP was first introduced as a GapP analog of BPP [10, 4]. It has a number of interesting properties.

1. AWPP is “robust” in the sense that its definition can be tweaked in several ways without changing the class [10, 7]. See Proposition 1.5, below.

2. AWPP is low for PP [10].

3. AWPP includes all of BQP [8] and is currently the smallest such counting class known.

4. There are relativized worlds where $P = AWPP \neq NP$ (among other things) [8].

The fourth property immediately gives a relativized world where $P = BQP \neq NP$, that is, where quantum computing is no more powerful than classical computing but NP complete problems are still hard (for both classical and quantum computation). This improves upon earlier results about random oracles and BQP [2].

A hint of the robustness of AWPP is given by the following proposition, which we state without proof. We will use this proposition in the proof of Theorem 4.2. The last equivalence below is actually the original definition of AWPP in [10].

**Proposition 1.5 ([10, 7])** Let $L$ be any language. The following are equivalent:

1. $L \in \text{AWPP}$. 

2. There are \( f, g \in \text{GapP} \) with \( g \) positive-valued such that for all \( x \in \Sigma^* \),

\[
x \in L \implies \frac{2}{3} \leq \frac{f(x)}{g(0^{\lvert x \rvert})} \leq 1,
\]

\[
x \notin L \implies 0 \leq \frac{f(x)}{g(0^{\lvert x \rvert})} \leq \frac{1}{3}.
\]

3. For every polynomial \( r \) there is a polynomial \( q \) and an \( f \in \text{GapP} \) such that for all \( n \in \mathbb{N} \) and \( x \in \Sigma^n \),

\[
x \in L \implies 1 - 2^{-r(n)} \leq 2^{-q(n)} f(x) \leq 1,
\]

\[
x \notin L \implies 0 \leq 2^{-q(n)} f(x) \leq 2^{-r(n)}.
\]

In the current paper we will propose classes similar to AWPP but where we restrict the relevant GapP function to be the result of matrix multiplications, and so these classes will all be included in AWPP. For this purpose, the property of GapP we will use the most is that it is closed under uniform multiplication of a polynomial number of matrices, each of which can have exponential size. Roughly speaking, if a small number of large matrices have entries computed by a GapP function, then their product’s entries are also computed by a GapP function. Since quantum operations are essentially matrix multiplication, this means we can simulate them with GapP, and so our proposed classes will all include BQP.

## 2 Matrices

Suppose \( a \) is a \( \mathbb{C} \)-valued function that takes two parameters \( i, j \in \mathbb{N} \) represented in binary, and possibly other parameters \( \vec{a} \). Then for numbers \( m, n \in \mathbb{N} \) (which may depend on \( \vec{a} \)) we let \( [a(\vec{a})]^{m \times n} \) denote the \( m \times n \) matrix whose \((i, j)\)th entry is \( a(\vec{a}; i, j) \), for \( 0 \leq i < m \) and \( 0 \leq j < n \). A lemma similar to the following was proved by Fortnow and Rogers [8] and also appears in [6].

**Lemma 2.1** Let \( a(\vec{a}, y; i, j) \) be a GapP function and let \( s(\vec{a}, y) \) be an FP function. Then there is a GapP function \( b(\vec{a}, y; i, j) \) such that for all \( \vec{a} \) and for all \( r \in \mathbb{N} \),

\[
[ b(\vec{a}, 1^r) ]^{s_r \times s_0} = [ a(\vec{a}, 1^r) ]^{s_r \times s_{r-1}} [ a(\vec{a}, 1^{r-1}) ]^{s_{r-1} \times s_{r-2}} \cdots [ a(\vec{a}, 1^1) ]^{s_1 \times s_0},
\]
where \( s_\ell = s(\overline{x}, 1^\ell) \) for \( 0 \leq \ell \leq r. \)

Proof. For \( 0 \leq i_r < s_r \) and \( 0 \leq i_0 < s_0 \), the \((i_r, i_0)\)th entry on the right hand side of the above equation is

\[
\sum_{s_1-1} \sum_{s_2-1} \cdots \sum_{s_{r-1}-1} \prod_{u=0}^{r} a(\overline{x}, u; i_u, i_{u-1}).
\]

This is a uniform exponential size sum of uniform polynomial size products of a GapP function. By the closure properties of GapP given in Proposition 1.3, it is a GapP function of \( \overline{x}, 1^r, i_r, \) and \( i_0. \)

\[\Box\]

3 Distribution-Valued Functions

Definition 3.1 A function \( f \) is a distribution-valued function (or DVF) if there is a polynomial \( p \) such that, for all \( n \in \mathbb{N} \) and \( x \in \Sigma^n \), \( p(n) \geq 1 \) and \( f(x) \) is a probability distribution on \( \Sigma^{p(n)} \). For \( y \in \Sigma^{p[n]} \), we write \( f(y \mid x) \) for the probability assigned to \( y \) by \( f(x) \). For every \( 0 \leq m < p(n) \), \( f(x) \) induces a natural probability distribution on \( \Sigma^m \) which assigns to each string \( z \in \Sigma^m \) the probability

\[
f(z \mid x) = \sum_{y \in \Sigma^{p(n)} : z \subseteq y} f(y \mid x).
\]

Note that \( f(\epsilon \mid x) = 1 \) for all \( x \).

FQP is a class of DVFs first defined in [1]. We give an essentially equivalent definition here. We assume here and throughout the paper that all gates in quantum circuits are drawn from a standard fixed finite universal family of quantum gates, such as \{\( H, T, H_1(\sigma_x) \)\} (see [11] for example).

Definition 3.2 ([1]) A DVF \( f \) is in FQP if and only if there is a polynomial \( p \) and a polynomial-time uniform family \( \{C_0, C_1, C_2, \ldots\} \) of quantum circuits with each \( C_n \) having \( p(n) \) inputs and the same number of outputs, such that for all \( n \), all \( x \in \Sigma^n \), and all \( y \in \Sigma^{p(n)} \), \( f(y \mid x) \) is the probability of observing \( |y\rangle \) as the output state of \( C_n \) where the input state is \( |x0^{p(n)-n}\rangle \).
In other words, if we start in the basis state corresponding to the string \( x \) being on the first \( n \) qubits and 0 on the rest, then \( f(y \mid x) \) is the probability of seeing the string \( y \) when we observe the output. Note that for strings \( z \) where \(|z| < |y| = p(n)\), \( f(z \mid x) \) is the probability of seeing \( z \) when we just measure the first \(|z|\) qubits of the output state.

**Definition 3.3** For any DVF \( f \), we define the language of \( f \), written \( L_f \), to be such that, for all \( x \in \Sigma^* \),

\[
x \in L_f \iff f(1 \mid x) > 1/2.
\]

We say that \( f \) has bounded error if for all \( x \in \Sigma^* \) and \( r \in \mathbb{N} \),

\[
f(1 \mid \langle x, 0^r \rangle) \leq 2^{-r} \quad \text{or} \quad f(1 \mid \langle x, 0^r \rangle) \geq 1 - 2^{-r}.
\]

**Definition 3.4** Let \( \mathcal{F} \) be a class of distribution-valued functions. We define the bounded error class of \( \mathcal{F} \) as

\[
B \cdot \mathcal{F} = \{ L_f : f \in \mathcal{F} \text{ has bounded error} \}.
\]

It is immediate from our definitions that \( B \cdot \text{FQP} = \text{BQP} \). We could, in an analogous way, introduce a class \( \text{FP} \) of DVFs defined by classical circuits with extra, random inputs; we would then get \( B \cdot \text{FP} = \text{BPP} \). The class \( \text{AWPP} \) itself can also be seen to be of the form \( B \cdot \mathcal{F} \) where \( \mathcal{F} \) is some appropriately concocted set of "normalized" GapP functions.

We wish to concentrate mostly on classes \( \mathcal{F} \) of DVFs defined using matrix multiplication and GapP functions. As mentioned above, such classes will yield

\[
\text{BQP} \subseteq B \cdot \mathcal{F} \subseteq \text{AWPP},
\]

and one of our goals is to investigate whether the second inclusion can be proper.

### 4 Classes of DVFs Defined by Matrices

For simplicity, we will restrict our attention to matrices over \( \mathbb{R} \). Generalizing to complex matrices is straightforward.

**Definition 4.1** Let \( \mathcal{C} \) be a class of functions \( \Sigma^* \to \mathbb{Z} \).
1. A DVF $f$ is in the class $1\text{-FM}(\mathcal{C})$ if and only if there is a polynomial $p \geq 1$ and a $g(x; i, j) \in \mathcal{C}$ such that for all $n \in \mathbb{N}$, all $x \in \Sigma^n$, and all $y \in \Sigma^{p(n)}$ we have

$$f(y \mid x) = \frac{(M_{y,x})^2}{|M_{x,x}|},$$

where

$$M = [g(0^n)]^{2^{p(n)} \times 2^n},$$

and $|M_{x,x}|$ (the Euclidean norm of $M_{x,x}$) is assumed to be nonzero.

2. The class $\text{poly-FM}(\mathcal{C})$ is defined similarly, except that

$$M = [g(0^n, 0^{\lfloor q(n) \rfloor})]^{2^{p(n)} \times 2^{p(n)}} \cdots [g(0^n, 0)]^{2^{p(n)} \times 2^{p(n)}} [g(0^n, \epsilon)]^{2^{p(n)} \times 2^n},$$

where $q$ is a polynomial chosen along with $p$.

This definition is a bit liberal in that we do not put any restrictions on the type of matrix $M$. In fact,

**Theorem 4.2**

$I\text{-FM}(\text{GapP}) = \text{poly-FM}(\text{GapP})$ and $B \cdot I\text{-FM}(\text{GapP}) = \text{AWPP}$.

Our matrix-based DVF classes will be based largely on Definition 4.1, but with restrictions placed on the type of matrix $M$.

Quantum circuits (before the measurement) correspond to unitary operators. So perhaps we should restrict our matrices to be unitary. There are two technicalities involved in this restriction, both resulting from all our matrix entries being integers: (i) we should allow the matrix to be normalized; (ii) we should allow the matrix to approximate a unitary matrix to arbitrary accuracy. Note also that for real matrices, unitary is the same as orthogonal.

**Definition 4.3** Let $\mathcal{C}$ be a class of functions $\Sigma^* \to \mathbb{Z}$.

1. A DVF $f$ is in the class $1\text{-FUM}(\mathcal{C})$ if and only if there is a polynomial $p$ and a $g(x; y; i, j) \in \mathcal{C}$ such that, for all $n \in \mathbb{N}$, $p(n) \geq n$ and there exists a $p(n) \times p(n)$ orthogonal matrix $U$ and a sequence of positive real numbers $d_0, d_1, d_2, \ldots$ such that

$$\left\| U - [g(0^n, 0^r)/d_r]^{p(n) \times p(n)} \right\| \leq 2^{-r}$$

for all $r \in \mathbb{N}$, and for all $x \in \Sigma^n$, and all $y \in \Sigma^{p(n)}$ we have

$$f(y \mid x) = (U_{y,x}^p)^2.$$
2. The class poly-FUM(C) is defined similarly, except that we approximate
U with matrices of the form
\[
\begin{bmatrix}
g(0^n, 0, 0) \\
g(0^n, 0, 0) \\
\vdots \\
g(0^n, 0, 0)
\end{bmatrix}^{2^p(n) \times 2^p(n)}
\]
for all \( r \in \mathbb{N} \), where \( q \) is a polynomial chosen along with \( p \).

**Theorem 4.4**

\( \text{FQP} \subseteq \text{poly-FUM(FP)} \subseteq \text{1-FUM(GapP)} = \text{poly-FUM(GapP)} \subseteq \text{1-FM(GapP)}. \)

**Corollary 4.5**

\( \text{BQP} \subseteq \text{B} \cdot \text{poly-FUM(FP)} \subseteq \text{B} \cdot \text{1-FUM(GapP)} \subseteq \text{AWPP}. \)

We suspect the last inclusion in Corollary 4.5 is proper.

Unitarity is a “global” condition of the matrix, which is difficult to enforce
if we are only computing each entry “locally.” As a consequence, it is un-
clear whether we can effectively enumerate, say, all 1-FUM(GapP) machines.
On the other hand, antisymmetry in matrices is easy to enforce locally, and
indeed we can effectively enumerate functions computing antisymmetric ma-
trices. This suggests an alternate approach: we make the matrix \( A \) being
computed by the \( C \) function to be antisymmetric, but we define \( f(y \mid x) \) in
terms of the matrix \( e^{A} \), which is orthogonal. This is analogous to specifying
a quantum operation by giving its Hamiltonian.

**Definition 4.6**

Let \( C \) be a class of functions \( \Sigma^* \rightarrow \mathbb{Z} \).

1. A DVF \( f \) is in the class 1-FAM(\( C \)) if and only if there is a polynomial
   \( p \), a function \( g(x; i, j) \in C \), and an FP function \( d : \Sigma^* \rightarrow \mathbb{N} \) such that,
   for all \( n \in \mathbb{N} \), \( p(n) \geq n \), \( d(0^n) > 0 \), and the matrix
   \[
   A = \begin{bmatrix} g(0^n) / d(0^n) \end{bmatrix}^{2^p(n) \times 2^p(n)}
   \]
is antisymmetric, and, for all \( x \in \Sigma^n \) and all \( y \in \Sigma^{p(n)} \) we have
   \[
   f(y \mid x) = (U_{y, x^{p(n)-n}})^2,
   \]
   where \( U = e^{A} \).

2. The class poly-FAM(\( C \)) is defined similarly, except that
   \[
   U = e^{A_1} e^{A_{q(n)-1}} \cdots e^{A_1},
   \]
   where \( q \) is a polynomial chosen along with \( p \) and
   \[
   A_j = \begin{bmatrix} g(0^n, 0^j) / d(0^n, 0^j) \end{bmatrix}^{2^p(n) \times 2^p(n)}
   \]
   for all \( 1 \leq j \leq q(n) \).
**Theorem 4.7** \( B \cdot 1\text{-FAM}(\text{GapP}) \subseteq B \cdot 1\text{-FUM}(\text{GapP}) \).

We conjecture that the two classes in Theorem 4.7 are equal.

**References**


11


