# Classical Simulation Complexity of Quantum Branching Programs

Farid Ablayev*
Kazan State University

**Abstract**

We present classical simulation techniques for measure once quantum branching programs.

For bounded error syntactic quantum branching program of width $w$ that computes a function with error $\delta$ we present a classical deterministic branching program of the same length and width at most $(1 + 2/(1 - 2\delta))^{2w}$ that computes the same function.

Second technique is a classical stochastic simulation technique for bounded error and unbounded error quantum branching programs. Our result is that it is possible stochastically-classically simulate quantum branching programs with the same length and almost the same width, but we lost bounded error acceptance property.

## 1 Introduction

Investigations of different aspects of quantum computations in the last decade became intensively growing area of mathematics, computer science, and physics. A good source of information on quantum computations is Nielsen's and Chuang's book [7]. The interest in models of quantum computation has been steadily increasing since the discovery of a polynomial time algorithm for factoring by Peter Shor [13]. During the last decade different types of quantum computation models based on Turing Machines, finite automata, circuits, and branching programs have been considered. For several of these models of computations, some examples of functions were presented for which quantum models appear to be much more efficient than their classical counterparts.

Complexity of classical simulation of quantum computations for different models of computations were investigated in numerous papers [4, 8, 11, 16, 12].

Branching programs are important model of computations, because of their natural relationships to machines models (Turing machines, automata) and Circuit models. Different restricted models of branching programs are widely used for hardware verification and in numerous CAD applications. In the paper we present two classical simulation techniques for measure once quantum branching programs.

*Work done in part while visiting Max-Plank Institute for Mathematics Bonn in 2007 Email: ablayev@ksu.ru

Our first result is the following. For bounded error syntactic quantum branching program [1] see also the journal version [3] of width $w$ that computes a function with error $\delta$ we present a classical deterministic branching program of the same length and width at most $(1 + 2/(1 - 2\delta))^{2w}$ that computes the same function. The construction of corresponding deterministic branching program is based on the following properties:

1. Quantum states are unit vectors (a set of quantum states form bounded set for $|| \cdot ||_2$ norm).

2. Unitary transformations of quantum states preserves a distance.

3. Bounded-error acceptance criteria.

Bounded-error acceptance criteria together with the properties 1 and 2 forms topological structure on the set of quantum states which leads to a desired deterministic branching program.

Second technique is a classical stochastic simulation technique for bounded-error quantum branching programs. Our result is that a resulting stochastic classical branching program is of the same length and almost the same width, but we lost bounded-error acceptance. Our construction of classical stochastic branching program is based on:

1. Replacing complex matrices with real ones with dimension doubled and tensor product construction as a bridge between $|| \cdot ||_1$ norm and $|| \cdot ||_2$ norm (Lemma 3). This construction gives new matrices with quadratic increase in dimension.

2. A Turakainen-type construction [14] to replace arbitrary real matrices with stochastic ones with "good properties" of the original ones (Lemma 4).

# 2 Definitions and Results

We start with definition of branching programs according to [15] (we call it *constructive* definition). Then we give an algebraic definition of branching programs and present results of the paper.

**Definition 1** *A* branching program *(BP) on the variable set $X = \{x_1, \ldots, x_n\}$ is a finite directed acyclic graph with one source node and sink nodes partitioned into two sets –* Accept *and* Reject*. Each non-sink node is labeled by a variable $x_i$ and has two outgoing edges labeled* 0 *and* 1 *respectively. An input $\sigma$ is* accepted *if and only if it induces a chain of transitions leading to a node in* Accept*, and the set of such inputs is the language accepted by the program.*

A branching program is *oblivious* if the nodes can be partitioned into levels $V_1, \ldots, V_\ell$ and a level $V_{\ell+1}$ such that the nodes in $V_{\ell+1}$ are the sink nodes, nodes in each level $V_j$ with $j \leq \ell$ have outgoing edges only to nodes in the next level $V_{j+1}$, and all nodes in a given level $V_j$ query the same bit $\sigma_{i_j}$ of the input.

**Definition 2** *The* size $Size(P)$ *of a branching program* $P$ *is the number of its non-sink nodes. The* length $Length(P)$ *of branching program* $P$ *is the maximum length of a path from the source to one of the sinks. The* width $Width(P)$ *of oblivious branching program* $P$ *is the number* $Width(P) = \max_j |V_j|$.

Clearly we have that length of branching program corresponds to time of computation and width on a level $j$ corresponds to a space that can be used on the step $j$ of computation.

In this paper we deal with polynomial size branching programs. Recall that arbitrary branching program can be transformed to oblivious branching program (see for example [6]) with only polynomial increasing the size. So without loss of generality we consider only oblivious branching programs in this paper.

Now we give a definition of a *linear branching program* based on oblivious model. This definition is a generalization of the definition of quantum branching program presented in [2]. Deterministic, stochastic, and quantum oblivious branching programs are particular cases of linear branching programs. Let $\mathbf{V}^k$ be a $k$-dimensional vector space. We use $|\mu\rangle$ and $\langle\mu|$ to denote column vectors and row vectors of $\mathbf{V}^k$, respectively, and $\langle\mu_1 \mid \mu_2\rangle$ denotes the complex inner product. We write $\mu$ when it is not important whether $\mu$ is a column or a row vector.

**Definition 3 (Linear branching program)** *A Linear Branching Program (LBP) P over* $\mathbf{V}^k$ *is defined as*

$$\mathcal{P} = \langle T, |\mu_0\rangle, \mathrm{Accept}\rangle \ ,$$

*where* $T = (T_1, \ldots, T_\ell)$ *is a sequence (of length* $\ell$*) of instructions. Each instruction* $T_j$ *is a triple* $T_j = \{i_j, M_j(0), M_j(1)\}$*, where* $i_j$ *determines a variable* $x_{i_j}$ *tested on the step* $j$*,* $M_j(0)$ *and* $M_j(1)$ *are* $k$*-dimensional linear transformations of the vector space* $\mathbf{V}^k$*. Vectors* $|\mu\rangle \in \mathbf{V}^k$ *are called states (state vectors) of* $P$*,* $|\mu_0\rangle \in \mathbf{V}^k$ *is the initial state of* $P$*, and* $\mathrm{Accept} \subseteq \{1, \ldots, k\}$ *is the accepting set.*

*According to the definition 2 it is natural to define the* $Width(P)$ *of linear BP as the* dimension *of state space* $\mathbf{V}^k$*. Further for LBP* $P$ *it is natural do define its size as* $Size(P) = Width(P)Length(P)$*.*

*We define a computation on* $P$ *with an input* $\sigma = \sigma_1 \ldots \sigma_n \in \{0,1\}^n$ *as follows:*

1. *A computation of* $P$ *starts from the initial state* $|\mu_0\rangle$*;*

2. *The j'th step of computation of* $P$ *applies instruction* $T_j$*: program* $P$ *queries a variable* $x_{i_j}$*, and applies the transition matrix* $M_j(\sigma_{i_j})$ *to the current state* $\mu$ *to obtain the state* $\mu' = M_j(\sigma_{i_j})\mu$*;*

3. *The final state (i.e., the state after step* $\ell$*) is*

$$|\mu(\sigma)\rangle = \prod_{j=\ell}^{1} M_j(\sigma_{i_j})|\mu_0\rangle \ .$$

Now oblivious deterministic, stochastic, and quantum branching programs can be presented as follows:

**Deterministic branching programs.** A *deterministic* branching program is a linear branching program over a vector space $\mathbb{F}^k$ for a corresponding finite field $\mathbb{F}$. A state $\mu$ of such a program is a vector with exactly one 1, the rest elemants are 0. The transition matrices $M$ have exactly one 1 in each column, the rest elements of the column are 0.

**Stochastic branching programs.** A *stochastic* branching program is a linear branching program over a real vector space $\mathbb{R}^k$. The concept of deterministic branching programs naturally generalizes to stochastic branching programs (SBP), by letting $\mu$ be a probability distribution, and by letting the $M_j$ be *stochastic* matrices, i.e., matrices with non-negative entries where each column sums to 1.

For a deterministic and stochastic branching program $P$, for an input $\sigma \in \{0,1\}^n$ we define the acceptance probability of $\sigma$ as follows

$$\Pr^P(\sigma) = Pr(\mu(\sigma)) = \sum_{i \in \text{Accept}} |\langle i \mid \mu(\sigma)\rangle| = \|\Pi_{\text{Accept}}\mu(\sigma)\|_1 \quad . \tag{1}$$

Here $|i\rangle$ is the basis vector with support on the node $i$ (unit vector with value 1 at $i$ and 0 elsewhere), and $\Pi_{\text{Accept}}$ is a projection operator on the *accepting subspace* span$\{|i\rangle : i \in \text{Accept}\}$.

**Quantum branching programs.** We define a *quantum* branching program (QBP) as a linear branching program over a Hilbert space $\mathcal{H}^k$. The $\mu$ for such a program are complex state vectors with $\|\mu\|_2 = 1$, and the $M_j$ are complex-valued unitary matrices. For a quantum branching program $P$, for an input $\sigma \in \{0,1\}^n$ we define the acceptance probability of $\sigma$ as follows

$$\Pr^P(\sigma) = Pr(\mu(\sigma)) = \sum_{i \in \text{Accept}} |\langle i \mid \mu(\sigma)\rangle|^2 = \|\Pi_{\text{Accept}}\mu(\sigma)\|_2^2 \,, \tag{2}$$

that is, the probability that if we measure $\mu(\sigma)$, we will observe it in the accepting subspace. Note that this is a "measure-once" model analogous to the model of quantum finite automata in [9], in which the system evolves unitarily except for a single measurement at the end.

Notice that in contrast to algebraic definition of quantum and stochastic BPs one can define these models in a (so called) *constructive* form. See for example book [15] for constructive definition of SBP and the paper [11] for constructive definition of QBP.

**Acceptance criteria.** We say that a LBP $P$ computes a Boolean function $f$ *with unbounded error* if $\Pr^P(\sigma) > 1/2$ if $f(\sigma) = 1$ and $\Pr^P(\sigma) \leq 1/2$ if $f(\sigma) = 0$. *We say that $P$ computes $f$ with threshold* $1/2$.

We say that a LBP $P$ computes a Boolean function $f$ *with bounded error* if there is some $\epsilon > 0$ such that $\Pr^P(\sigma) \geq 1/2 + \epsilon$ if $f(\sigma) = 1$ and $\Pr^P(\sigma) \leq 1/2 - \epsilon$ if $f(\sigma) = 0$. *We say that $P$ computes $f$ with error* $\delta = 1/2 - \epsilon$ (with margin $\epsilon$).

## 2.1 Deterministic Simulations of Stochastic and Quantum Branching Programs

**Syntactic Stochastic and Quantum Programs** For unbounded and bounded error stochastic and quantum branching programs we define two subsets $\mathcal{A}$ and $\mathcal{R}$ of the set $\mathcal{F}$ of sink state vectors (consistent and inconsistent) as follows. For unbounded error programs, we define

$$\mathcal{A} = \{\mu \in \mathcal{V}_{\ell+1} : \Pr(\mu) > 1/2\} \quad \text{and} \quad \mathcal{R} = \{\mu \in \mathcal{V}_{\ell+1} : \Pr(\mu) \leq 1/2\};$$

and for bounded error programs, we define

$$\mathcal{A} = \{\mu \in \mathcal{V}_{\ell+1} : \Pr(\mu) \geq 1/2 + \epsilon\} \quad \text{and} \quad \mathcal{R} = \{\mu \in \mathcal{V}_{\ell+1} : \Pr(\mu) \leq 1/2 - \epsilon\}$$

We call $\mathcal{A}$ and $\mathcal{R}$ the *accepting* and *rejecting* sets respectively.

Recall that $\mathcal{V}_{\ell+1}$ includes the final states reachable by all possible paths, both consistent and inconsistent. Then:

**Definition 4 ([1])** *We call a stochastic or a quantum branching program* syntactic *if its accepting and rejecting set of state vectors form a partition of the set of sink states, i.e., if $\mathcal{V}_{\ell+1} = \mathcal{A} \cup \mathcal{R}$.*

Note that without the syntactic restriction, it might happen that $\mathcal{V}_{\ell+1} \neq \mathcal{A} \cup \mathcal{R}$, and that some inconsistent final state vector $\mu \in \mathcal{V}_{\ell+1}$ has the property that $1/2 - \epsilon < \Pr(\mu) < 1/2 + \epsilon$.

**Theorem 1 (Deterministic Simulation Theorem)** *Let function $f$ be bounded error $\delta$ ($\delta \in (0, 1/2)$) computed by syntactic QBP $P$. Then there exists deterministic BP $P'$ that computes $f$ and has the following complexity characteristics: $Length(P') = Length(P)$ and*

$$Width(P') \leq \left(1 + \frac{2}{1 - 2\delta}\right)^{2Width(P)}.$$

Similarly, we can deterministically simulate classical stochastic BP.

**Theorem 2** *If a function is computed with bounded error $\delta$ ($\delta \in (0, 1/2)$) by a width-$w$ syntactic stochastic branching program, then it is also computed by a deterministic branching program of the same length, and width*

$$w' \leq \left(1 + \frac{2}{1 - 2\delta}\right)^{w}.$$

The proof of Theorem 1 and Theorem 2 is in the section Proofs.

Notice that Theorem 1 and Theorem 2 imply that constant width quantum and stochastic branching programs can be simulated by a constant width deterministic branching programs and hence — by an $NC^1$ circuits. For more information and results see [1].

## 2.2 Stochastic Classical Simulation of Quantum Branching Programs

**Theorem 3 (Stochastic Simulation Theorem)** *Let function $f$ be unbounded error (bounded error) computed by QBP $Q$. Then there exists SBP $P$ that unbounded error computes $f$ of the same length $Length(P) = Length(Q)$ and width $Width(P) = 4Width^2(Q) + 3$.*

We present the proof of Theorem 3 in the section Proofs.

Now we define probabilistic and quantum complexity classes based on branching programs as follows.

**Definition 5** *Let BPP-BP and PP-BP be the classes of functions computable with bounded error and unbounded error respectively by stochastic branching program of polynomial size;*

*Let BQP-BP and PrQP-BP be the classes of functions computable with bounded error and unbounded error respectively by quantum branching program of polynomial size.*

**Theorem 4**

$$PrQP\text{-}BP \subseteq PP\text{-}BP$$

$$BQP\text{-}BP \subseteq PP\text{-}BP$$

*Proof.* The proof of the Theorem is the consequence of the Simulation Theorem 3. □

Notice that Sasaki in [12] proved that $BQP\text{-}BP \subseteq BPL/poly$ where $BPL/poly$ is a class of languages accepted by Logarithmic-space bounded error nonuniform probabilistic Turing machines.

# 3 Proofs

## 3.1 Proof of Theorem 1 and Theorem 2

We start with the idea of lower bounds proof and notations we use for formal proof.

Let us denote $(w,l)$-$P$ an $w$ width and $l$ length $BP$. The idea of proofs of Theorem 1 and Theorem 2 is that having syntactic stochastic (quantum) $(w,l)$-$P$ that computes a function $f$ with margin $\epsilon$ we construct a deterministic $BP$ $(w',l)$ - $P'$ such that $P'$ computes the same function $f$ and

$$w' \leq \left(1 + \frac{1}{\epsilon}\right)^{w}$$

when $P$ is stochastic $BP$ and

$$w' \leq \left(1 + \frac{1}{\epsilon}\right)^{2w}$$

when $P$ is quantum $BP$.

The construction of $P'$ is based on the following properties. We will view on computation by oblivious (stochastic and quantum) $BP$ $(w,l)$-$P$ as an $l$ step linear process of transformation of vectors $\alpha$ of a current internal description ($ID$) of $P$. In stochastic case $\alpha = \mu$, where $\mu = (p_1, \ldots, p_w)$ is a current probability distribution of a states of $P$, and in quantum case $\alpha = |\psi\rangle$, where $|\psi\rangle = (z_1, \ldots, z_w)^T$ is a current distribution of amplitudes of states of $P$.

We represent this $l$ step linear process as an $(l+1)$-leveled deterministic branching program $DP$, as follows: each node of $DP$ labeled by vector $\alpha$ corresponds to an $ID$ of $P$, and each level $i \in \{0, \ldots, l\}$ represents a step of the computation. The level 0 contains initial node labeled $\alpha_0$ — the initial distribution of $P$. From each node $\alpha$ on the level $i$, $i \in \{0, \ldots, l-1\}$, out goes two edges labeled $x_{j_i} = 0$ and $x_{j_i} = 1$ where $x_{j_i}$ is a variable tested in the computational step $i$. Edge $x_{j_i} = 1$, directed from parent $\alpha$ on a level $i$ to child $\alpha'$ on the level $i+1$ iff $P$ being on the step $i$ of computation in $ID$ $\alpha$ tests $x_{j_i} = 1$ and transforms its $ID$ on the step $i+1$ to $\alpha'$.

Denote *Accept* (*Reject*) a set of all accepting (rejecting) sink nodes of $P$. Sink nodes of $DP$ on the level $l$ are labeled in addition by 1 ("accept") and 0 ("reject") as follows: in stochastic case sink node $\alpha = (p_1, \ldots, p_w)$ labeled 1 if $Pr(\alpha) = \sum_{j \in Accept} p_j \geq 1/2 + \epsilon$, and labeled 0 if $Pr(\alpha) = \sum_{j \in Accept} p_j \leq 1/2 - \epsilon$; in quantum case node $\alpha = (z_1, \ldots, z_w)^T$ labeled 1 if $Pr(\alpha) = \sum_{j \in Accept} |z_j|^2 \geq 1/2 + \epsilon$, and labeled 0 if $Pr(\alpha) = \sum_{j \in Accept} |z_j|^2 \leq 1/2 - \epsilon$. Denote $\mathcal{A}$ ($\mathcal{R}$) a set of all sink nodes of $DP$ labeled 1 (0).

¿From the above we have the following property.

**Property 1** *Deterministic branching program $DP$ computes the same Boolean function $f$ as $P$.*

In the next section we use metric point of view to $DP$ for constructing deterministic $BP$ $(w', l)$-$P'$ that computes the same function as $DP$ and hence the same function as $P$.

**Metric Properties of** $DP$ Denote $\Psi_i$, $i \in \{0, \ldots, l\}$, a set all possible $ID$s of $P$ on step $i$ of computation. Let $\Psi = \cup_{i=0}^{l} \Psi_i$. For stochastic $P$ we define metric on the space $\Psi$ based on norm $||\cdot||_1$. That is, for $\alpha = (p_1, \ldots, p_w)$ and $\alpha' = (p'_1, \ldots, p'_w)$ $\rho(\alpha, \alpha') = ||\alpha - \alpha'||_1 = \sum_{i=1}^{w} |p_i - p'_i|$. For quantum $P$ we define metric on the space $\Psi$ based on norm $||\cdot||_2$: for $\alpha = (z_1, \ldots, z_w)$ and $\alpha' = (z'_1, \ldots, z'_w)$ $\rho(\alpha, \alpha') = ||\alpha - \alpha||_2$. We will also use notation $||\cdot||$ for norm $||\cdot||_2$.

Recall known notions of metric spaces we need in the proof (see for example [5]). Let $\mathbf{V}^w$ be an $w$-dimensional vector space (real valued or complex valued) with metric $\rho$. Points $\mu, \mu'$ from $\mathbf{V}^w$ are connected through $\theta$-chain if there exists a finite set of points $\mu_1, \mu_2, \ldots, \mu_m$ from $\mathbf{V}^w$ such that $\mu_1 = \mu, \mu_m = \mu'$ and $\rho(\mu_i, \mu_{i+1}) < \theta$ for $i \in \{1, \ldots, m-1\}$. For $\mathbf{V}^w$ its subset $C$ is called $\theta$-component if arbitrary two points $\mu, \mu' \in \mathcal{C}$ are connected through $\theta$-chain. It is known [5] that if $\mathcal{D}$ is a finite diameter subset of a subspace of $\mathbf{V}^w$ (diameter of $\mathcal{D}$ is defined as $\sup_{\mu,\mu' \in \mathcal{D}} \{\rho(\mu, \mu')\}$ then for $\theta > 0$ $\mathcal{D}$ is partitioned to a finite number $t$ of its $\theta$-components.

**Lemma 1** *Let $f$ be a Boolean function $(1/2+\epsilon)$-computed by $P$. Let $DP$ be a corresponding deterministic $BP$ for $P$. Let $\theta > 0$ and let for sink nodes of $DP$ the following holds: for arbitrary $\alpha \in \mathcal{A}$ and $\alpha' \in \mathcal{R}$ it is holds that $\rho(\alpha, \alpha') \geq \theta$. Then, there exists a deterministic $BP$ $(w', l)$-$P$ which computes $f$ and*

$$w' \leq \left(1 + \frac{2}{\theta}\right)^{w}$$

*when $P$ is stochastic $BP$, and*

$$w' \leq \left(1 + \frac{2}{\theta}\right)^{2w}$$

*when $P$ is quantum $BP$.*

*Proof.* Consider $\mathcal{D}$ a sphere of radius 1 with center in $(0,\dots,0)$. We clearly have that $\Psi \subseteq \mathcal{D}$. From the condition of the lemma it follows that subsets $\mathcal{A}$ and $\mathcal{R}$ of $\Psi_l$ is a union of some $\theta$-components of $\Psi_l$. Next. Oblivious property of $BP$ $P$ provides the following: on the each level $i$, $i \in \{0,\dots,l-1\}$, of $DP$ for all nodes $\alpha \in \Psi_i$ it is tested the same variable $x_{j_i}$ and applied the same linear transition $M_{x_{j_i}}(1)$ if $x_{j_i}=1$ ($M_{x_{j_i}}(0)$ if $x_{j_i}=0$) where $M_{x_{j_i}}(a)$ is stochastic matrix when $P$ is stochastic $BP$ and $M_{x_{j_i}}(a)$ is unitary when $P$ is quantum $BP$.

It is known (and it can be easily verified) that transformations determined by stochastic matrix $M$ does not increase the distance. That is, if $\alpha' = M\alpha$ and $\beta' = M\beta$ then $\rho(\alpha',\beta') \le \rho(\alpha,\beta)$. Unitary matrix $M$ preserves the distance. That is, it is holds that $||\alpha'-\beta'|| = ||\alpha-\beta||$.

Denote $\mathcal{C}_i$ the set of all $\theta$-components of $\Psi_i$. For $C \in \mathcal{C}_i$ and matrix $M$ we denote $MC = \{\alpha' : \alpha = M\alpha, \alpha \in C\}$. From the property of non increasing distance linear transformations (stochastic or uniform) it is holds that for $C \in \mathcal{C}_i$ and for $a \in \{0,1\}$ there exists $C' \in \mathcal{C}_{i+1}$ such that $MC \subseteq C'$ for the stochastic $P$ and $MC = C'$ for the quantum $P$.

Now we describe deterministic $BP$ $P'$ that computes $f$ as follows: $P'$ is an $l$-leveled oblivious $BP$. On the level $i$ tested variable $x_{j_i}$ (as in $BP$) and all nodes are labeled by $\theta$-components $C \in \mathcal{C}_i$. From the node $C \in \mathcal{C}_i$ edge labeled $x_{j_i} = a$ goes to to node $C' \in \mathcal{C}_{i+1}$ iff $M_{x_{j_i}}(a)C \subseteq C'$.

¿From the above it follows that $P'$ computes the same function as $DP$

The width $w(P')$ of $P'$ is $w' = \max\{t_0,\dots,t_l\}$ where $t_i$ is the number $\theta$-components of $\Psi_i$. Let $w' = t_i$. We estimate the number $t$ of $\theta$-components (number of nodes of $B$) of $\Psi_j$ as follows.

For each $\theta$-component $C \in \mathcal{C}_i$ select one point $\alpha \in C$. If we draw a sphere of the radius $\theta/2$ with the center $\alpha \in C$ then all such spheres do not intersect pairwise. All these $w'$ spheres are in large sphere of radius $1+\theta/2$ which has center $(0,0,\dots,0)$. The volume of a sphere of a radius $r$ in $\mathcal{H}_w$ is $cr^w$ when $\mathcal{H}_w$ is real space and is $cr^{2w}$ when $\mathcal{H}_w$ is complex space (in the complex space $\mathcal{H}_w$ each complex point is a 2-dimensional point). Constant $c$ depends on the metric of $\mathcal{H}_w$. Now for stochastic and quantum case we have respectively that

$$w' \le \frac{c\,(1+\theta/2)^w}{c\,(\theta/2)^w} = \left(1+\frac{2}{\theta}\right)^w, \qquad w' \le \frac{c\,(1+\theta/2)^{2w}}{c\,(\theta/2)^{2w}} = \left(1+\frac{2}{\theta}\right)^{2w},$$

□

Below we present technical lemma that estimates parameter $\theta$ of the lemma 1 depending on margin $\epsilon$ of computation.

**Lemma 2** *Let $f$ be a Boolean function ($1/2+\epsilon$)-computed by $(w,l)$-$P$. Let $DP$ be a corresponding deterministic $BP$ for $P$.*

*Then for arbitrary $\alpha \in \mathcal{A}$ and $\alpha' \in \mathcal{R}$ for the case of stochastic $P$ it is holds that:*

$$||\alpha-\alpha'||_1 \ge \theta = 2\epsilon,$$

*for the case of quantum $P$ it is holds that:*

$$||\alpha-\alpha'|| \ge \theta = 2\epsilon.$$

*Proof.* Consider the case of stochastic $P$. $\alpha = (p_1 \dots, p_w)^T$, $\alpha' = (p'_1 \dots, p'_w)^T$,

$$||\alpha - \alpha'||_1 = \sum_{i=1}^{w} |p_i - p'_i| \geq \sum_{i \in Accept} |p_i - p'_i| \geq |\sum_{i \in Accept} p_i - \sum_{i \in Accept} p'_i|$$

From the condition of the lemma we have that $\sum_{i \in Accept} p_i \geq 1/2 + \epsilon$ and $\sum_{i \in Accept} p'_i \leq 1/2 - \epsilon$. From this we have that

$$||\alpha - \alpha'||_1 \geq 1/2 + \epsilon - (1/2 - \epsilon) = 2\epsilon.$$

Consider the case of quantum $P$. $\alpha = (z_1, \dots, z_d)^T$ and $\alpha' = (z'_1, \dots z'_d)^T$.

¿From the condition of the lemma it is holds that

$$2\epsilon \leq \sum_{i \in Accept} (|z_i|^2 - |z'_i|^2) = \sum_{i \in Accept} (|z_i| - |z'_i|)(|z_i| + |z'_i|) \leq \sum_{i \in Accept} (|z_i - z'_i|)(|z_i| + |z'_i|)$$

and similarly

$$2\epsilon \leq \sum_{i \in Reject} (|z'_i|^2 - |z_i|^2) = \sum_{i \in Reject} (|z'_i| - |z_i|)(|z_i| + |z'_i|) \leq \sum_{i \in Reject} (|z_i - z'_i|)(|z_i| + |z'_i|)$$

or

$$4\epsilon \leq \sum_{i=1}^{w} (|z_i - z'_i|)(|z_i| + |z'_i|).$$

¿From the above using known inequality $\sum_{i=1}^{d} a_i b_i \leq \sqrt{\sum_{i=1}^{d} a_i^2}\sqrt{\sum_{i=1}^{d} b_i^2}$ and triangle inequality for the norm we get that

$$4\epsilon \leq ||\alpha - \alpha'||(||\,|\alpha| + |\alpha'|\,|| \leq ||\alpha - \alpha'||(||\alpha|| + ||\alpha'|| = 2||\alpha - \alpha'||$$

□

Now the lower bounds of Theorem 1 and Theorem 2 follows immediately from lemmas 1 and 2. This completes the proof of theorem 1 □

## 3.2 Proof of Stochastic Simulation Theorem 3

We call LBP $P$ a program of Type I, if it uses metric (1) when defining the acceptance probability, and a program of Type II, if metric (2) is used.

The proof is based on three lemmas we present below.

**Lemma 3** *Let function $f$ be computed by QBP $Q$. Then there exists LBP $P$ of Type I that computes $f$ with $Width(P) = 4Width(Q)^2$ and $Length(P) = Length(Q)$ such that $Pr^Q(\sigma) = Pr^P(\sigma)$ for each $\sigma \in \{0,1\}^n$.*

*Proof sketch.* First using the known real-valued simulation of complex-valued matrix multiplication from QBP $Q$ over $\mathbf{C}^k$ we construct LBP $P'$ of Type II over $\mathbf{R}^{2k}$ with $Width(P') = 2Width(Q)$ and $Length(P') = Length(Q)$ such that $Pr^{P'}(\sigma) = Pr^{Q}(\sigma)$ for each $\sigma \in \{0,1\}^n$ (see [2] for more details ).

Next we construct LBP $P$ of Type I from LBP $P'$ of Type II with $Width(P) = Width(P')^2$ and $Length(P) = Length(P')$ such that $Pr^{P'}(\sigma) = Pr^{P}(\sigma)$ for each $\sigma \in \{0,1\}^n$. Here we use relation among LBP of Type I and Type II (among "linear" and "non linear" extracting a result of computation) used in [9]. For completeness of presentation we display it here for branching programs.

Let $d = 2k$. Let LBP $P' = \langle T, |\mu_0\rangle, \mathrm{Accept}\rangle$ where $T = (\{i_j, M_j(0), M_j(1)\})_{j=1}^{\ell}$. We construct $P = \langle T', |\tau_0\rangle, \mathrm{Accept}'\rangle$ as follows. The initial state $|\tau_0\rangle = |\mu_0 \otimes \mu_0\rangle$ — is $d^2$-dimension vector, $T' = (\{i_j, W_j(0), W_j(1)\})_{j=1}^{\ell}$ where $W_j(\sigma) = M_j(\sigma) \otimes M_j(\sigma)$ is $d^2 \times d^2$ matrix. Accepting set $\mathrm{Accept}' \subseteq \{1, \ldots, d^2\}$ of states is defined according to *Accept* $\subseteq \{1, \ldots, d\}$ as follows $\mathrm{Accept}' = \{j : j = (i-1)d + i,\, i \in \mathrm{Accept}\}$.

Using the fact that for real valued vectors $c$, $b$ it holds that $\langle c|b\rangle^2 = \langle c \otimes c | b \otimes b\rangle$ we have that $\prod_{j=\ell}^{1} W_j(\sigma_{i_j}) = \prod_{j=\ell}^{1}(M_j(\sigma_{i_j}) \otimes M_j(\sigma_{i_j})) = \prod_{j=\ell}^{1} M_j(\sigma_{i_j}) \otimes \prod_{j=\ell}^{1} M_j(\sigma_{i_j})$.

$$
\begin{aligned}
Pr^{P}(\sigma) &= \sum_{i \in F'} \langle i| \prod_{j=\ell}^{1} W_j(\sigma_{i_j}) |\tau_0\rangle = \sum_{i \in F} \langle i \otimes i| \prod_{j=\ell}^{1} (M_j(\sigma_{i_j}) \otimes M_j(\sigma_{i_j})) |\mu_0 \otimes \mu_0\rangle \\
&= \sum_{i \in F} \langle i| \prod_{j=\ell}^{1} M_j(\sigma_{i_j}) |\mu_0\rangle^2 = Pr^{P'}(\sigma).
\end{aligned}
$$

¿From the construction of LBP $P$ we have that $Width(P) = 4Width(Q)^2$ and $Length(P) = Length(Q)$. $\square$

**Lemma 4** *Let function $f$ be computed by LBP $P$ of Type I. Then there exists SBP $P'$ that computes $f$ with $Width(P') = Width(P) + 2$, $Length(P') = Length(P)$ such that for each $\sigma = \{0,1\}^n$ it is true that*

$$Pr^{P'}(\sigma) = c^{\ell} Pr^{P}(\sigma) + 1/(d+2)$$

*where $\ell = Length(P)$, $d = Width(P)$ , constant $c \in (0,1]$ depends on program $P$.*

*Proof sketch.* Let $P = \langle T, |\mu_0\rangle, \mathrm{Accept}\rangle$, where $T = (T_1, \ldots, T_\ell)$ and $T_j = \{i_j, M_j(0), M_j(1)\}$. Without loss of generality we suppose that a set Accept consists only of one node. One can easily construct LBP with unique accepting node from LBP with several accepting nodes (without increasing width and length) using standard technique from Linear Automata Theory (see for example [10, 14]).

Let $d = Width(P)$. We construct SBP $P' = \langle T', |\mu_0'\rangle, \mathrm{Accept}'\rangle$ as follows. For each instruction $T_j$ of program $P$ we define instruction $T_j' = \{i_j, W_j(0), W_j(1)\}$ of $P'$ as follows.

First for each $d \times d$ matrix $M$ of instruction $T_j$ we define $(d+2) \times (d+2)$ matrix

$$A = \left( \begin{array}{c|c|c} 0 & 0 \ldots 0 & 0 \\ \hline b & M & \vdots \\ \hline \beta & q & 0 \end{array} \right),$$

such that sum of elements of each row and each column of $A$ is zero (we are free to select elements of column $b$, row $q$ and number $\beta$).

Matrix $A$ has the property: sum of elements of each row and each column of $A$ is zero. It is easy to verify that product of such matrices also would be a matrix of the such type.

Now let $R$ be stochastic $(d+2) \times (d+2)$ matrix who's $(i,j)$-entry is $1/(d+2)$. Select positive constant $c \leq 1$ such that matrix $W$, defined as

$$W = cA + R$$

is stochastic matrix. Further by induction on $\ell$ we have that the product of matrices of type $W$ is also stochastic matrix of the same structure. Now for an input $\sigma = \sigma_1 \ldots \sigma_n$ we have that

$$W(\sigma) = \prod_{j=\ell}^{1} W_j(\sigma_{i_j}) = c^\ell \prod_{j=\ell}^{1} A_j(\sigma_{i_j}) + R.$$

By selecting suitable initial probabilities distribution $|\mu_0'\rangle$ and accepting nodes we can pick up from $W(\sigma)$ entry we need (entry that gives $\sigma$ accepting probability). From the construction of SBP $P'$ we have that $Width(P') = Width(P) + 2$, $Length(P') = Length(P)$, $Pr^{P'}(\sigma) = c^\ell Pr^P(\sigma) + 1/(d+2)$ for each $\sigma = \{0,1\}^n$. □

Lemma 4 says that having Type I LBP $P$ that process its input $\sigma$ with threshold $1/2$ one can construct SBP $P'$ that process $\sigma$ with threshold $\lambda = c^\ell 1/2 + 1/(d+2)$, where $\ell = Length(P)$ and $d = Width(P)$.

**Lemma 5** *Let SBP $P$ computes $f$ with threshold $\lambda \in [0,1)$. Then for arbitrary $\lambda' \in (\lambda, 1)$ there exists SBP $P$ that computes $f$ with threshold $\lambda'$ such that $Width(P') = Width(P) + 1$, $Length(P') = Length(P)$.*

*Proof:* The proof uses standard technique from Probabilistic Automata Theory (see for example the book [10]) and is omitted. □

Lemmas 3,4,5 prove the statement of Theorem 3.

# References


[1] F. Ablayev, C. Moore, and C. Pollett, Quantum and Stochastic Branching Programs of Bounded Width, *Electronic Colloquium on Computational Complexity*, TR02-013, 2002, available at `http://www.eccc.uni-trier.de/eccc/`

[2] F. Ablayev, A. Gainutdinova, and M. Karpinski. On computational Power of quantum branching programs. *Proc. FCT 2001*, Lecture Notes in Computer Science 2138: 59–70, 2001.

[3] F. Ablayev, A. Gainutdinova, M. Karpinski, C. Moore, C. Pollett, *On the computational power of probabilistic and quantum branching program* Inf. Comput. 203(2): 145-162 (2005)

[4] L. Adleman, J. Demarrais, M. Huang. Quantum computability, SIAM J. on Computing. 26(5), 1997, 1524–1540.

[5] P. Alexandrov. Introduction to set theory and general topology. Berlin, 1984.

[6] D. Barrington. Bounded-width polynomial branching programs recognize exactly those languages in $NC^1$. *Journal of Computer and System Sciences* 38(1): 150–164, 1989.

[7] M. Nielson and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press. 2000.

[8] L. Fortnow. One complexity theorist's view of quantum computing. *Theoretical Computer Science*, 2003, 292(3), 597–610.

[9] C. Moore, J. Crutchfield. Quantum Automata and Quantum Grammars. *Theoretical Computer Science,* 2000, 237, 275–306.

[10] A. Paz. Introduction to Probabilistic Automata. Academic-Press, 1971.

[11] M. Sauerhoff and D. Sieling. Quantum branching programs and space-bounded nonuniform quantum complexity. *ph/0403164, March 2004.*

[12] Y. Sasaki. Nonunifrom Quantum Complexity: Bounded-error quantum branching programs and their computational power. *Poster of Tokyo Office Quantum Computation Seminar*, 2002.

[13] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* 26(5): 1484–1509, 1997.

[14] P. Turakainen. Generalized automata and stochastic languages, *Proc. of AMS 21*, 1969, 303–309.

[15] I. Wegener. *Branching Programs and Binary Decision Diagrams.* SIAM Monographs on Discrete Mathematics and Applications. 2000.

[16] J. Watrous. On quantum and classical space bounded processes with algebraic transition amplitudes. *Proc. of FOCS 1999*, 341-351.