Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 85 (2008)

On Complexity of Quantum Branching Programs Computing Equality-like Boolean Functions

Farid Ablayev Airat Khasianov Alexander Vasiliev

January 25, 2010

Abstract

We consider the *Hidden Subgroup*, and *Equality*-related problems in the context of quantum *Ordered Binary Decision Diagrams*. For the decision versions of considered problems we show polynomial upper bounds in terms of quantum OBDD width. We apply a new modification of the fingerprinting technique and present the algorithms in circuit notation. Our algorithms require at most *logarithmic* number of qubits.

1 Introduction

Considering one-way quantum finite automata, Ambainis and Freivalds (see [AF98]) suggested that first quantum-mechanical computers would consist of a comparatively simple quantum-mechanical part connected to a classical computer. In this paper we consider another restricted model of quantum-classical computation referred to as *oblivious Ordered Read-Once Quantum Branching Programs*. It is also known as non-uniform automata.

Two models of *quantum branching programs* were introduced by Ablayev, Gainutdinova, Karpinski [AGK01] (*leveled programs*), and by Nakanishi, Hamaguchi, Kashiwabara [NHK00] (*non-leveled programs*). Later it was shown by Sauerhoff [SS04] that these two models are polynomially equivalent.

In this paper we use the generalized *fingerprinting* technique introduced in [AV08]. The basic ideas of this approach date back to 1979 [Fre79] (see also [MR95]). It was later successfully applied in the *quantum automata* setting by Ambainis and Freivald in 1998 [AF98] (later improved in [AN08]). Subsequently, the same technique was adapted for the quantum branching programs by Ablayev, Gainutdinova and Karpinski in 2001 [AGK01], and was later generalized in [AV08].

The *hidden subgroup problem* [ME99], [Høy97] is an important computational problem that has factoring and discrete logarithm as its special cases. Subsequently, an efficient algorithm for the hidden subgroup problem implies efficient solutions for both the *period finding problem*, and original *Simon problem*.

We show refined proof of the linear upper bound for the *Hidden Subgroup Problem* [KH06]. We prove linear upper bounds for *Equality*, *Palindrome* and boolean variant of *Periodicity* and *Semi-Simon* problems. Our upper bounds hold for arbitrary *ordering* of the input variables and were initially presented in [KH05], and can also be found in [AKK].

ISSN 1433-8092

2 Preliminaries and Definitions

The definition of a *linear branching program* is a generalization of the definition of quantum branching program presented in [AGK01]. Deterministic and quantum oblivious branching programs are special cases of linear branching programs. Let $\mathbf{V}^d$ be a d-dimensional vector space. We use $|\psi\rangle$ and $\langle\psi|$ to denote column vectors and row vectors respectively from $\mathbf{V}^d$, and $\langle\psi_1 \mid \psi_2\rangle$ denotes the inner product.

Definition 1 (Linear branching program)**.** *A Linear Branching Program P of width d and length l (a $(d,l)-LBP$) over $\mathbf{V}^d$ is defined as*

$$P = \langle T, |\psi_0\rangle, \text{Accept}\rangle$$

where T is a sequence of l instructions: $T_j = \big(x_{i_j}, U_j(0), U_j(1)\big)$ depends on x_{i_j} tested on the step j, and $U_j(0)$, $U_j(1)$ are $d \times d$ matrices.

Vectors $|\psi\rangle \in \mathbf{V}^d$ are called states (state vectors) of P, $|\psi_0\rangle \in \mathbf{V}^d$ is the initial state of P, and Accept $\subseteq \{1,\ldots,d\}$ *is the accepting set.*

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

1. *A computation of P starts from the initial state $|\psi_0\rangle$;*
2. *The j'th instruction of P queries a variable x_{i_j}, and applies the transition matrix $U_j = U_j(\sigma_{i_j})$ to the current state $|\psi\rangle$ to obtain the state $|\psi'\rangle = U_j(\sigma_{i_j})\,|\psi\rangle$;*
3. *The final state is*

$$|\psi(\sigma)\rangle = \left(\prod_{j=l}^{1} U_j(\sigma_{i_j})\right)|\psi_0\rangle \quad .$$

The usual complexity measures for $(d,l)-LBP$ are its width d, length l, and size $d \cdot l$.

Deterministic branching programs. A *deterministic* branching program is a linear branching program over a vector space $\mathbb{R}^d$. A state $|\psi\rangle$ of such a program is a Boolean vector with exactly one 1. The matrices U_j correspond to permutations of order d, and so have exactly one 1 in each column. For branching programs over groups this is true for the rows as well; in which case, the U_j are permutation matrices.

Quantum branching programs. We define a *quantum* branching program as a linear branching program over a Hilbert space $\mathcal{H}^d$. The $|\psi\rangle$ for such a program are complex state vectors with $\|\,|\psi\rangle\,\|_2 = 1$, and the U_j are complex-valued unitary matrices.

After the l-th (last) step of quantum transformation P measures its configuration $|\psi_\sigma\rangle$ where $|\psi_\sigma\rangle = U_l(\sigma_{i_l})U_{l-1}(\sigma_{i_{l-1}})\ldots U_1(\sigma_{i_1})\,|\psi_0\rangle$. Measurement is presented by a diagonal zero-one projection matrix M where $M_{ii} = 1$ if $i \in Accept$ and $M_{ii} = 0$ if $i \notin Accept$. The probability $Pr_{accept}(\sigma)$ of P accepting input σ is defined by

$$Pr_{accept}(\sigma) = ||M\,|\psi_\sigma\rangle\,||^2.$$

Bounded error computation. A QBP P *computes the Boolean function* f *with margin* $\epsilon \in (0, 1/2)$ if for all inputs the probability of error is bounded by $1/2 - \epsilon$.

A QBP P *computes* f *with one-sided error* if there exists an $\epsilon > 0$ such that for all $\sigma \in f^{-1}(1)$ the probability of P accepting σ is 1 and for all $\sigma \in f^{-1}(0)$ the probability of P accepting σ is less than ϵ.

Note that this is a "measure-once" model analogous to the model of quantum finite automata in [MC97], in which the system evolves unitarily except for a single measurement at the end. We could also allow multiple measurements during the computation, by representing the state as a density matrix ρ, and by making the U_j superoperators, but we do not consider it here.

Read-once branching programs.

Definition 2. *We call an* LBP P *an* OBDD *or read-once* LBP *if each variable* $x \in \{x_1, \dots, x_n\}$ *occurs in the sequence* T *of transformations of* P *at most once.*

The "obliviousness" is inherent for an LBP and therefore this definition is consistent with the usual notion of an OBDD. We will use QOBDD for quantum read-once branching programs and OBDD for deterministic ones.

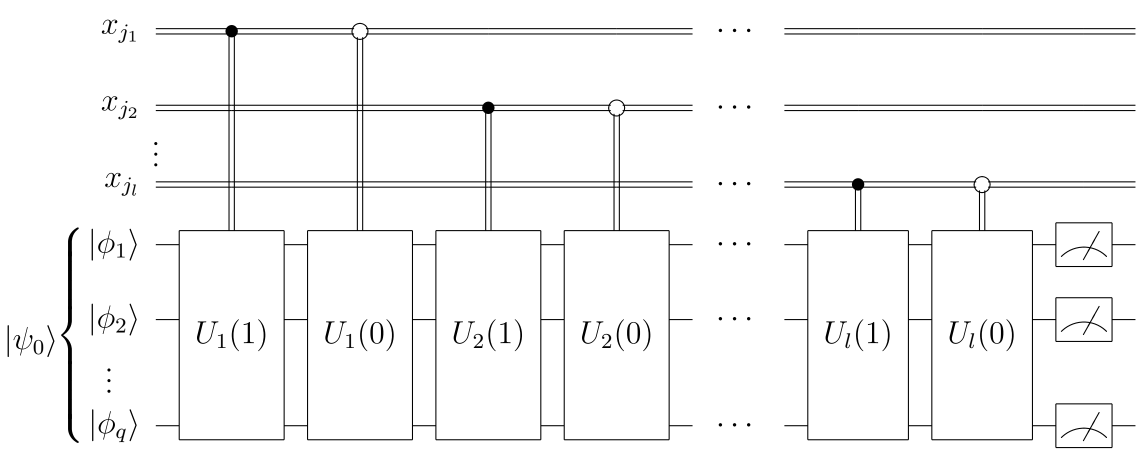
The following general lower bound on the width of QOBDDs is proven in [AGK01].

Theorem 1. *Let* $\epsilon \in (0, 1/2)$. *Let* $f(x_1, \dots, x_n)$ *be a Boolean function* $(1/2+\epsilon)$*-computed (computed with margin* ϵ*) by a quantum read-once branching program* Q. *Then*

$$\mathrm{width}(Q) = \Omega\left(\frac{\log \mathrm{width}(P)}{2\log\left(1+\frac{1}{\epsilon}\right)}\right)$$

where P *is a deterministic OBDD of minimal width computing* $f(x_1, \dots, x_n)$.

Circuit representation. A QBP can be viewed as a quantum circuit aided with an ability to read classical bits as control variables for unitary operations. That is any quantum circuit is a QBP which does not depend essentially on it's classical inputs.



Here $x_{j_1}, \dots, x_{j_l}$ is the sequence of (not necessarily distinct) variables denoting classical control bits.

Note that for a QBP in the circuit setting another important complexity measure explicitly comes out – a number of qubits q physically needed to implement a corresponding quantum system with classical control. From definition it follows that $\log d \le q \le d/2$. The maximum of $d/2$ is reached when all the qubits do not interfere and thus are isolated quantum systems.

Definition 3. *We call a (d, l)-QBP P a q-qubit QBP if the program P can be implemented as a classically controlled quantum system based on q qubits.*

3 Quantum Fingerprinting

Fingerprinting is the technique that allows presenting objects (words over some finite alphabet) by their *fingerprints*, which are significantly smaller than the originals. Moreover, they are intended to reliably extract the important information about the input with one-sided error. In this paper we use the *fingerprinting* technique developed in [AV08].

Our approach has the following properties:

- It is designed for models with classical control and thus for QBPs.
- Fingerprints are easy to create, we use only controlled rotations about the same axis of the Bloch sphere by the similar angle and Hadamard gates (for more information see, e.g. [NC00]).
- The lemma we prove guarantees the existence of a "good" set of parameters which allows to obtain the ϵ upper bound on the error probability, where ϵ is a constant, and $0 < \epsilon < 1$.

Fingerprinting technique For the problem being solved we choose some cardinal m, an error probability bound $\epsilon > 0$, fix $t = \lceil (2/\epsilon) \ln 2m \rceil$, and construct a mapping $g : \{0, 1\}^n \to \mathbb{Z}$. Then for arbitrary binary string $\sigma = \sigma_1 \dots \sigma_n$ we create it's fingerprint $|h_\sigma\rangle$ composing t single qubit fingerprints $|h^i_\sigma\rangle$:

$$\begin{aligned} |h^i_\sigma\rangle &= \cos \tfrac{2\pi k_i g(\sigma)}{m} |0\rangle + \sin \tfrac{2\pi k_i g(\sigma)}{m} |1\rangle \\ |h_\sigma\rangle &= \tfrac{1}{\sqrt{t}} \sum_{i=1}^{t} |i\rangle |h^i_\sigma\rangle \end{aligned}$$

That is, the last qubit is rotated by t different angles about the $\hat{y}$ axis of the Bloch sphere.

The chosen parameters $k_i \in \{1, \dots, m-1\}$ for $i = \overline{1, t}$ are "good" in the following sense.

Definition 4. *A set of parameters $K = \{k_1, \dots, k_t\}$ is called "good" for $g \neq 0 \bmod m$ if*

$$\frac{1}{t^2} \left(\sum_{i=1}^{t} \cos \frac{2\pi k_i g}{m} \right)^2 < \epsilon.$$

The left side of inequality is the squared amplitude of the basis state $|0\rangle^{\otimes \log t} |0\rangle$ if the operator $H^{\otimes \log t} \otimes I$ has been applied to the fingerprint $|h_\sigma\rangle$. Informally, that kind of set guarantees, that the probability of error will be bounded by a constant below 1.

The following lemma proves the existence of a "good" set and follows the proof of the corresponding statement from [AN08].

Lemma 1. *There is a set K with $|K| = t = \lceil (2/\epsilon) \ln 2m \rceil$ which is "good" for all $g \neq 0 \bmod m$.*

Proof. Using Azuma's inequality (see, e.g., [MR95]) we prove that a random choice of the set K is "good" with positive probability .

Let $1 \leq g \leq m-1$ and let K be the set of t parameters selected uniformly at random from $\{0, \dots, m-1\}$.

We define random variables $X_i = \cos\frac{2\pi k_i g}{m}$ and $Y_k = \sum_{i=1}^{k} X_i$. We want to prove that Azuma's inequality is applicable to the sequence $Y_0 = 0$, Y_1, Y_2, Y_3, …, i.e. it is a martingale with bounded differences. First, we need to prove that $E[Y_k] < \infty$.

From the definition of X_i it follows that

$$E[X_i] = \frac{1}{m}\sum_{j=0}^{m-1}\cos\frac{2\pi jg}{m}$$

Consider the following weighted sum of m-th roots of unity

$$\frac{1}{m}\sum_{j=0}^{m-1}\exp\left(\frac{2\pi jg}{m}i\right) = \frac{1}{m}\cdot\frac{\exp(2\pi igm/m)-1}{\exp(2\pi ig/m)-1} = 0,$$

since g is not a multiple of m.

$E[X_i]$ is exactly the real part of the previous sum and thus is equal to 0.

Consequently, $E[Y_k] = \sum_{i=1}^{k} E[X_i] = 0 < \infty$.

Second, we need to show that the conditional expected value of the next observation, given all the past observations, is equal to the last observation.

$$E[Y_{k+1}|Y_1,\ldots,Y_k] = \frac{1}{m}\sum_{j=0}^{m-1}\left(Y_k + \cos\frac{2\pi jg}{m}\right) = Y_k + \frac{1}{m}\sum_{j=0}^{m-1}\cos\frac{2\pi jg}{m} = Y_k$$

Since $|Y_{k+1} - Y_k| = |X_{k+1}| \le 1$ for $k \ge 0$ we apply Azuma's inequality to obtain

$$Pr(|Y_t - Y_0| \ge \lambda) = Pr\left(|\sum_{i=1}^{t} X_i| \ge \lambda\right) \le 2\exp\left(-\frac{\lambda^2}{2t}\right)$$

Therefore, we induce that the probability of K being not "good" for $1 \le g \le m-1$ is at most

$$Pr\left(|\sum_{i=1}^{t} X_i| \ge \sqrt{\epsilon}t\right) \le 2\exp\left(-\frac{\epsilon t}{2}\right) \le \frac{1}{m}$$

for $t = \lceil (2/\epsilon)\ln 2m\rceil$.

Hence the probability that constructed set is not "good" for at least one $1 \le g \le m-1$ is at most $(m-1)/m < 1$. Therefore, there exists a set which is "good" for all $1 \le g \le m-1$. This set will also be "good" for all $g \neq 0 \bmod m$ because $\cos\frac{2\pi k(g+jm)}{m} = \cos\frac{2\pi kg}{m}$. □

We use this result for our fingerprinting technique choosing the set $K = \{k_1,\ldots,k_t\}$ which is "good" for all $g = g(\sigma) \neq 0$. That is, it allows to distinguish those inputs whose image is 0 modulo m from the others.

That hints on how this technique may be applied:

1. We construct $g(x)$, that maps all acceptable inputs to 0 modulo m and others to arbitrary non-zero (modulo m) integers.

2. After the necessary manipulations with the fingerprint the $H^{\otimes \log t}$ operator is applied to the first $\log t$ qubits. This operation "collects" all of the cosine amplitudes at the all-zero state. That is, we obtain the state of type

$$|h'_\sigma\rangle = \frac{1}{t}\sum_{i=1}^{t}\cos\left(\frac{2\pi k_i g(\sigma)}{m}\right)|00\ldots0\rangle|0\rangle + \sum_{i=2}^{2t}\alpha_i|i\rangle$$

3. Then this state is measured in the standard computational basis and we accept the input if the outcome is the all-zero state. This happens with probability

$$Pr_{accept}(\sigma) = \frac{1}{t^2}\left(\sum_{i=1}^{t} \cos\frac{2\pi k_i g(\sigma)}{m}\right)^2,$$

which is 1 for inputs, whose image is 0 mod m, and is bounded by ϵ for the others.

4 The Upper Bounds for Some Boolean Functions

4.1 Equality

We shall first demonstrate our approach on *Equality* function.

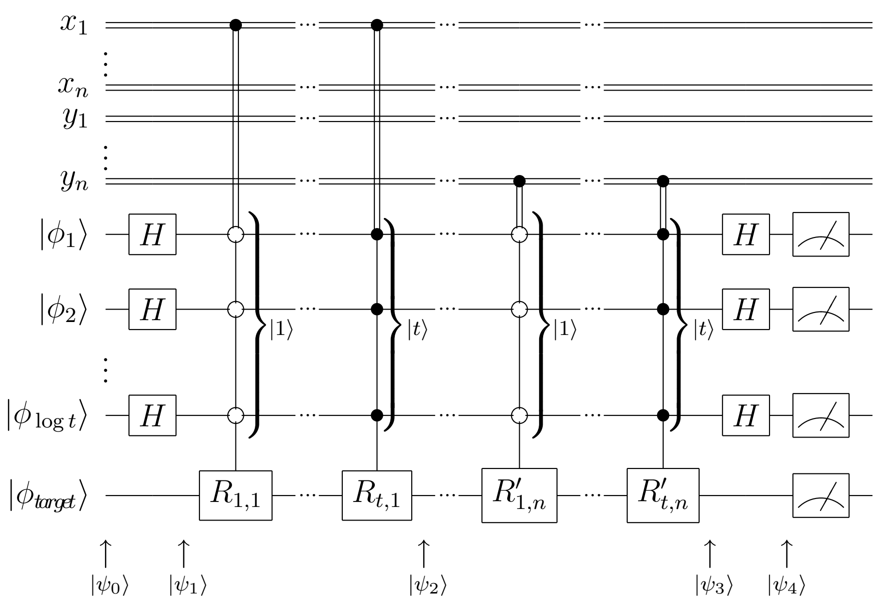
Definition 5. *$EQ_n(x, y) \equiv [x = y]$, where n is even, and $x = \{x_1, \ldots, x_{n/2}\}$, $y = \{x_{n/2+1}, \ldots, x_n\}$.*

This function is easy in deterministic case for a clever choice of the variable ordering. But for the natural ordering, we consider here, it is exponentially hard.

Theorem 2. *For arbitrary $\epsilon \in (0, 1)$ the function $EQ_n(x, y)$ can be computed with one-sided error ϵ by a QOBDD of width $O(n)$, where $n = |xy|$ is the length of the input.*

Proof. For the assignment of the x part of the input we introduce the notation σ_x. In the similar sense, we introduce the notation σ_y. We shall use these notations throughout the proof.

We present our algorithm in circuit notation.



Initially $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_{\log t}\rangle \otimes |\psi_{target}\rangle = |00\ldots0\rangle$. For $i \in \{1, \ldots, t\}$ we define rotations $R_{i,j}$ as follows

$$R_{i,j} = \begin{cases} R_{\hat{y}}\left(\frac{4\pi k_i 2^{n/2-j}}{2^{n/2}}\right) & \text{for } j \le n/2 \\ R_{\hat{y}}\left(-\frac{4\pi k_i 2^{n-j}}{2^{n/2}}\right) & \text{for } j > n/2 \end{cases},$$

and the set of parameters $K = \{k_1, \ldots, k_t\}$ is "good" according to the Definition 4 with $t = \lceil (2/\epsilon) \ln(2 \cdot 2^{n/2}) \rceil$.

The state of the system after having read the input σ is

$$|\psi\rangle \quad = \quad \frac{1}{\sqrt{t}}\sum_{i=1}^{t}|i\rangle\left(\cos\frac{2\pi k_i(\sigma_x-\sigma_y)}{2^{n/2}}|0\rangle+\sin\frac{2\pi k_i(\sigma_x-\sigma_y)}{2^{n/2}}|1\rangle\right)$$

Applying $H^{\otimes \log t}\otimes I$ we obtain the state

$$|\psi'\rangle = \frac{1}{t}\sum_{i=1}^{t}\cos\left(\frac{2\pi k_i(\sigma_x-\sigma_y)}{2^{n/2}}\right)|00\dots0\rangle|0\rangle+\sum_{i=2}^{2t}\alpha_i|i\rangle,$$

where α_i – are some unimportant amplitudes.

The input σ is accepted if the measurement outcome of $|\psi_1\rangle\dots|\psi_{\log t}\rangle|\psi_{target}\rangle$ is $|00\dots0\rangle|0\rangle$. Clearly, the accepting probability is

$$Pr_{accept}(\sigma)=\frac{1}{t^2}\left(\sum_{i=1}^{t}\cos\frac{2\pi k_i(\sigma_x-\sigma_y)}{2^{n/2}}\right)^2$$

If $\sigma_x=\sigma_y$ then the program accepts σ with probability 1. Otherwise, we chose the set $K=\{k_1,\dots,k_t\}$ so that

$$Pr_{accept}(\sigma)=\frac{1}{t^2}\left(\sum_{i=1}^{t}\cos\frac{2\pi k_i(\sigma_x-\sigma_y)}{2^{n/2}}\right)^2<\epsilon$$

Now it is easy to see that we have used the *fingerprinting* technique from the section 3 with parameters $m=2^{n/2}$ and $g(x,y)=x-y$. Therefore, $EQ_n(x,y)$ can be computed by a $\log 2t$-qubit QOBDD, where $\log 2t=O(\log\log m)=O(\log n)$. □

4.2 Palindrome

Definition 6. $Palindrome_n(x_1,\dots,x_n)\equiv\left[x_1x_2\dots x_{\lfloor n/2\rfloor}=x_nx_{n-1}\dots x_{\lceil n/2\rceil+1}\right]$

As the corollary we can prove that $Palindrome_n$ has the the same complexity as $EQ_n(x,y)$.

Theorem 3. *For arbitrary $\epsilon\in(0,1)$ the function $Palindrome_n$ can be computed with constant one-sided error ϵ by a QOBDD of width $O(n)$.*

Proof. The proof of this result mimics the proof for $EQ_n(x,y)$, the only difference is the definition of $R_{i,j}$:

$$R_{i,j}=\begin{cases} R_{\hat{y}}\left(\frac{4\pi k_i 2^{\lfloor n/2\rfloor-j}}{2^{\lfloor n/2\rfloor}}\right) & \text{for } j\le\lfloor n/2\rfloor\\ R_{\hat{y}}\left(-\frac{4\pi k_i 2^{j-\lceil n/2\rceil-1}}{2^{\lfloor n/2\rfloor}}\right) & \text{for } j\ge\lceil n/2\rceil+1\end{cases}$$

□

4.3 Periodicity

For a set of input variables $x=\{x_0,\dots,x_{n-1}\}$, and s – the period parameter, we define the *Periodicity* function $Period_n^s(x)$ that takes the input of length $n+k$, where $n=|x|$, and $k=\lceil\log n\rceil$ – the number of bits needed for s.

$$Period_n^s(x)\equiv\begin{cases}1 & \text{if } x_i=x_{i+s \mod n}, i=\overline{0,n-1};\\ 0 & \text{otherwise.}\end{cases}$$

Theorem 4. *For arbitrary $\epsilon \in (0,1)$ the function $Period_n^s(x)$ can be computed with constant one-sided error ϵ by a QOBDD of width $O(n)$, where $n = |x|$.*

Proof. We use the algorithm for $EQ_n(x, y)$ with rotations

$$R_{i,j} = R_{\hat{y}}\left(\frac{4\pi k_i(2^j - 2^{j-s \bmod n})}{2^n}\right)$$

□

4.4 Semi-Simon

For a set of input variables $x = \{x_0, \ldots, x_{n-1}\}$, and $s \in (0, n]$ we define the *Semi-Simon* function as follows

$$\textit{Semi-Simon}_n^s(x) \equiv \begin{cases} 1 & x_i = x_{i \oplus s}, i = \overline{0, n-1}; \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\oplus$ is a bitwise addition modulo 2. Here we treat i both ways: as a natural number, and as a binary sequence representing the number.

Remark 1. *The way we treated binary sequences in the definition above, we should adopt throughout the paper without further notice.*

Theorem 5. *For any $\epsilon \in (0,1)$ and all $s \in (0, n]$ the function Semi-Simon$_n^s(x)$ can be computed with one-sided error ϵ by a QOBDD of width $O(n)$.*

Proof. Computing of equality function will be again in the core for the proof. We use rotations

$$R_{i,j} = R_{\hat{y}}\left(\frac{4\pi k_i(2^j - 2^{j \oplus s})}{2^n}\right)$$

□

4.5 Permutation Matrix Test Function

The *Permutation Matrix* test function ($PERM_n$) is defined on n^2 variables x_{ij} $(1 \leq i, j \leq n)$. It tests whether the input matrix contains exactly one 1 in each row and each column. Thus, $PERM_n = 1$ iff the input matrix contains exactly one 1 in each row and each column.

Note, that this function cannot be effectively computed by a deterministic OBDD – the lower bound is $\Omega(2^n n^{-5/2})$ regardless of the variable ordering [Weg00]. The width of the best known probabilistic OBDD, computing this function with one-sided error, is $O(n^4 \log n)$ [Weg00]. Our algorithm has the width $O(n \log n)$. Since the lower bound $\Omega(n - \log n)$ follows from Theorem 1, our algorithm is almost optimal.

Theorem 6. *For any $\epsilon \in (0,1)$ the function $PERM_n(x)$ can be computed with one-sided error ϵ by a QOBDD of width $O(n \log n)$.*

Proof. We introduce the notation $r_i = \sum_{j=1}^{n} x_{ij}$ for the number of ones in the i-th row, and $c_j = \sum_{i=1}^{n} x_{ij}$ for the number of ones in the j-th column. It's obvious, that $0 \leq r_i, c_j \leq n$, and $PERM_n(x_{11}, \ldots, x_{nn}) = 1$ iff all of r_i and c_j equal 1.

We encode all of these numbers as a $(n+1)$-ary number

$$N(x) = \sum_{i=1}^{n} r_i(n+1)^{i-1} + \sum_{j=1}^{n} c_j(n+1)^{n+j-1},$$

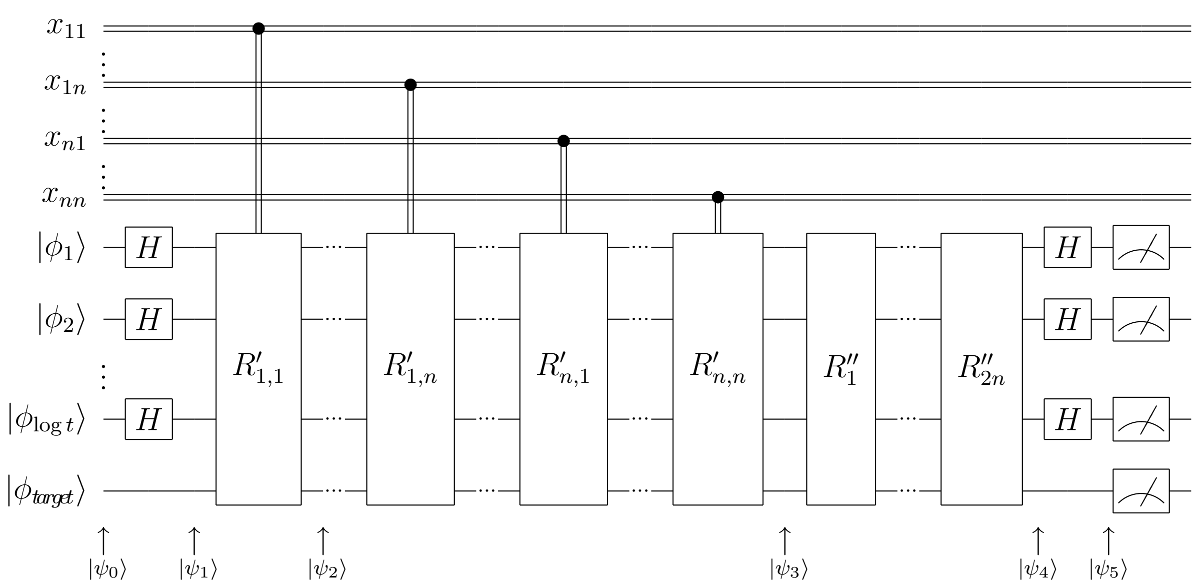
and denote

$$N_1 = \sum_{i=1}^{n}(n+1)^{i-1} + \sum_{j=1}^{n}(n+1)^{n+j-1} = \sum_{i=1}^{2n}(n+1)^{i-1}.$$

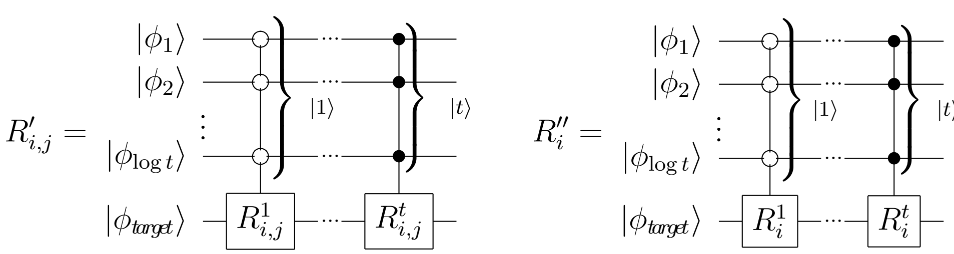
Therefore, all of r_i and c_j equal 1 iff $N(x) = N_1$ or, equivalently, when $g(x) = N(x) - N_1 = 0$.

We induce, that $PERM_n(x)$ can be computed by checking whether the input sequence has the property, encoded by $g(x)$, via the fingerprinting method.

Here is the algorithm in the circuit notation:



Initially qubits $|\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_{\log t}\rangle \otimes |\phi_{target}\rangle$ are in the state $|\psi_0\rangle = |0\rangle^{\otimes \log t}|0\rangle$. We define controlled unitary transformations $R'_{i,j}$ for $i, j \in \{1, \ldots, n\}$ and R''_i for $i \in \{1, \ldots, 2n\}$ by the following circuits



Here

$$\begin{aligned} R^l_{i,j} &= R_{\hat{y}}\left(\tfrac{4\pi k_l[(n+1)^{i-1}+(n+1)^{n+j-1}]}{(n+1)^{2n}}\right), \\ R^l_i &= R_{\hat{y}}\left(-\tfrac{4\pi k_l(n+1)^{i-1}}{(n+1)^{2n}}\right), \end{aligned}$$

and the set of parameters $K = \{k_1, \ldots, k_t\}$ is "good" according to the Definition 4 with $t = 2^{\lceil \log((2/\epsilon)\ln 2\cdot(n+1)^{2n})\rceil} = O(n \log n)$.

Note, that operator $R^l_{i,j}$ "appends" 1 to positions $i-1$ and $n+j-1$, corresponding to r_i and c_j. The transformation R^l_i rotates the qubit in the opposite direction by an angle, proportional to the number with 1 in the $(i-1)$-th position and 0 elsewhere.

Let $\sigma = \sigma_{11} \ldots \sigma_{nn} \in \{0,1\}^{n^2}$ be an input string.

The first layer of Hadamard operators transforms the state $|\psi_0\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_{\log t}\rangle$ into

$$|\psi_1\rangle = \frac{1}{\sqrt{t}} \sum_{l=1}^{t} |l\rangle |0\rangle .$$

Next, upon input symbol 0 identity transformation I is applied. But if the value of x_{ij} is 1, then the state of the last qubit is transformed by the operator $R^l_{i,j}$, rotating it by the angle $\frac{4\pi k_l[(n+1)^{i-1}+(n+1)^{n+j-1}]}{(n+1)^{2n}}$. Moreover, the rotation is done in each of t subspaces with the corresponding amplitude $1/\sqrt{t}$. Thus, the qubit $|\phi_{target}\rangle$ is in some sense simultaneously rotated by t similar angles about the $\hat{y}$ axis of the Bloch sphere. Such a parallelism is implemented by the controlled operators $C_l(R^l_{i,j})$, which transform the states $|l\rangle |\cdot\rangle$ into $|l\rangle R^l_{i,j} |\cdot\rangle$, and leave others unchanged. For instance, having read the input symbol $x_{11} = 1$, the system would evolve into state

$$|\psi_2\rangle \;=\; \frac{1}{\sqrt{t}} \sum_{l=1}^{t} |l\rangle \left(\cos \frac{2\pi k_l[(n+1)+(n+1)^n]}{(n+1)^{2n}} |0\rangle + \sin \frac{2\pi k_l[(n+1)+(n+1)^n]}{(n+1)^{2n}} |1\rangle \right) .$$

Thus, after having read the input σ the amplitudes would "collect" the value $N(\sigma)$

$$|\psi_3\rangle \;=\; \frac{1}{\sqrt{t}} \sum_{l=1}^{t} |l\rangle \left(\cos \frac{2\pi k_l N(\sigma)}{(n+1)^{2n}} |0\rangle + \sin \frac{2\pi k_l N(\sigma)}{(n+1)^{2n}} |1\rangle \right) .$$

At the next step we perform the rotations by the angle $-\frac{4\pi k_l N_1}{(n+1)^{2n}}$ about the $\hat{y}$ axis of the Bloch sphere for each $l \in \{1, \ldots, t\}$. Therefore, the state of the system would be

$$\begin{aligned} |\psi_4\rangle &= \frac{1}{\sqrt{t}} \sum_{l=1}^{t} |l\rangle \left(\cos \frac{2\pi k_l (N(\sigma)-N_1)}{(n+1)^{2n}} |0\rangle + \sin \frac{2\pi k_l (N(\sigma)-N_1)}{(n+1)^{2n}} |1\rangle \right) \\ &= \frac{1}{\sqrt{t}} \sum_{l=1}^{t} |l\rangle \left(\cos \frac{2\pi k_l g(\sigma)}{(n+1)^{2n}} |0\rangle + \sin \frac{2\pi k_l g(\sigma)}{(n+1)^{2n}} |1\rangle \right) , \end{aligned}$$

where $g(\sigma) = N(\sigma) - N_1$ checks whether σ defines a permutational matrix.

Applying $H^{\otimes \log t} \otimes I$ we obtain the state

$$\begin{aligned} |\psi_5\rangle &= \left(\frac{1}{t} \sum_{l=1}^{t} \cos \frac{2\pi k_l g(\sigma)}{(n+1)^{2n}} \right) |0\rangle^{\otimes \log t} |0\rangle + \\ &+ \;\; \gamma |0\rangle^{\otimes \log t} |1\rangle + \sum_{l=2}^{t} |l\rangle (\alpha_l |0\rangle + \beta_l |1\rangle) , \end{aligned}$$

where γ, α_i, and β_i are some unimportant amplitudes.

The input σ is accepted if the measurement outcome is $|0\rangle^{\otimes \log t} |0\rangle$. Clearly, the accepting probability is

$$Pr_{accept}(\sigma) = \frac{1}{t^2} \left(\sum_{l=1}^{t} \cos \frac{2\pi k_l g(\sigma)}{(n+1)^{2n}} \right)^2 .$$

$PERM_n(\sigma) = 1$ iff $g(\sigma) = 0$, and in that case the accepting probability is 1. Otherwise $g(\sigma) \neq 0$, and the probability of erroneously accepting $\sigma \in PERM_n^{-1}(0)$ is bounded by ϵ because of the choice of the set $K = \{k_1, \ldots, k_t\}$:

$$Pr_{accept}(\sigma) = \frac{1}{t^2} \left(\sum_{l=1}^{t} \cos \frac{2\pi k_l g(\sigma)}{(n+1)^{2n}} \right)^2 < \epsilon.$$

Thus, f can be computed by a q-qubit quantum OBDD, where $q = \log t + 1 = O(\log n + \log\log n)$. The width of the program is $2^q = 2t = O\left(\frac{n \log n}{\epsilon}\right)$, where ϵ is a constant, $0 < \epsilon < 1$. $\square$

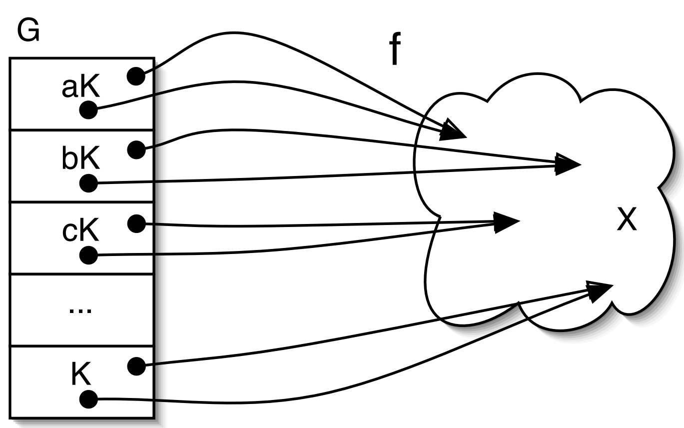
5 The upper bound for Hidden Subgroup Function

This problem was first defined and considered in [KH05]. The proof of the theorem in this section follows somewhat different presentation from [KH06]. In this paper we give a shorter and more illustrative proof of the result via circuit presentation, the approach first applied in [AV08].

In order to investigate Quantum Branching Program complexity of the *Hidden Subgroup Problem*, we define a function.

Definition 7. *Let K be a normal subgroup of a finite group G. Let X be a finite set. For a sequence $\chi \in X^{|G|}$ let $\sigma = bin(\chi)$ be its representation in binary. If σ encodes no correct sequence $\chi = \chi_1 \ldots \chi_{|X|}$, then* Hidden Subgroup *function of σ is set to be zero, otherwise:*

$$HSP_{G,K}(\sigma) = \begin{cases} 1, & if \quad \forall a \in G \ \forall\, i, j \in aK \ (\chi_i = \chi_j) \\ & and \quad \forall\, a, b \in G \ \forall\, i \in aK \ \forall\, j \in bK \\ & (aK \neq bK \Rightarrow \chi_i \neq \chi_j); \\ 0, & otherwise. \end{cases}$$



Let f be the function encoded by the input sequence. We want to know if a function $f : G \to X$ "hides" the subgroup K in the group G. Our program receives G and K as *parameters*, and function f as an *input string* containing values of f it takes on G. The values are arranged in lexicographical order. See Definition 7.

We make two assumptions. First, we assume that the set X contains exactly $(G : K)$ elements. Indeed, having read the function f, encoded in the input sequence σ, we have X to be the set of all different values that f takes. Obviously, if $|X|$ is less or greater than $(G : K)$, then $\mathrm{HSP}_{G,K}(\sigma) = 0$.

The second assumption, is that we replace all values of f by numbers from 1 through $(G:K)$. Thus, $\mathrm{HSP}_{G,K}(x_1,\ldots,x_n)$ is a Boolean function of $n=|G|\lceil\log G:K\rceil$ variables. In these two assumptions the following theorem holds.

Theorem 7. *Function $HSP_{G,K}(x)$ can be computed with one-sided error by a quantum OBDD of width $O(n)$.*

5.1 Proof of theorem 7

First we shall prove the following lemma.

Lemma 2. *In order to correctly compute $HSP_{G,K}(x)$ it is enough to perform following calculations.*

1. *For every coset we check equalities for all input sequence values that have indices from this coset;*
2. *From every coset we choose a representative, and check if the sum of values of f on all the representatives equals to the following value*

$$S=\sum_{i=1}^{G:K} i=\frac{(G:K)((G:K)+1)}{2}.$$

Proof. One direction is straightforward. The other direction is also not difficult. Suppose we have the two conditions of the lemma satisfied. Let aK and bK be two different cosets with elements $d\in aK$ and $c\in bK$, such that $\sigma_d=\sigma_c$, where σ_d, σ_c are binary encoded images $f(d)$ and $f(c)$ respectively. We fix $c\in bK$. There are two cases possible:

1. For all $d\in aK(\sigma_d=\sigma_c)$;
2. There exists $d'\in aK(\sigma_d\neq\sigma_c)$.

Apparently in the first case we indeed could choose any of the elements of a coset to check inequalities. In the second case the first condition of the lemma would fail. The reasoning for bK is analogous.

When the values of f are different on different cosets, obviously, the sum of these values is the sum of numbers from 1 through $G:K$. Therefore, $\mathrm{HSP}_{G,K}(\sigma)=1$ iff both conditions of the lemma are satisfied. ∎

In the plan laid down by the previous lemma, our algorithm will consist of two parts checking conditions of the lemma.

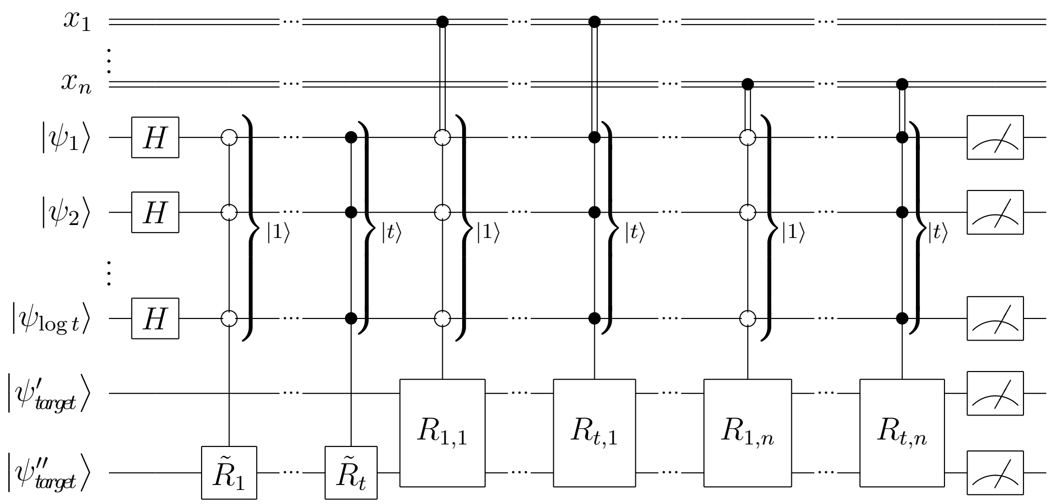
Additionally, we shall use another indexation of χ when convenient: $\chi_{a,q}$ is a value of f on the qth element of the coset aK.

Therefore, for a binary input symbol σ_j we define

- $a=a(j)$ for the number of the corresponding coset;
- We also define $a^{-1}(a_0)$ to be the index of the first bit of the first encountered element of a^{th} coset in the binary input sequence σ;
- $q=q(j)$ for the number of the corresponding element of the coset a;

- $r = r(j)$ for the number of bit in the binary representation of $\chi_{a,q}$

and start indexation from 0. Thus $a = \overline{0, (G:K)-1}, q = \overline{0, |aK|-1}$.



Denote $l = \lceil \log G : K \rceil$. First part of the algorithm performs calculation of the sequence of equalities:

$$\sigma_{a^{-1}(0)} = \sigma_{a^{-1}(1)} = \ldots = \sigma_{a^{-1}((G:K)-1)}$$

This can be done using the algorithm for Equality function with rotations:

$$R'_{i,j} \quad = \quad R_{\hat{y}} \left(\frac{2\pi k_i 2^{|K|al}(2^{ql+r} - 2^{(q+1)l+r \bmod |K|l})}{2^n} \right)$$

In this equation

- $ql + r$ is the index of the first, as it is appeared in the input sequence, representative of the current considered by the equality algorithm coset.
- Subsequently, $(q+1)l+r$ is the index of the corresponding bit in the binary representation of the first element of the next considered coset, to be compared with the currently considered bit of the binary input.
- We use the power expression modulo $|K|l$ in order to compare the first and the last cosets.

The second part of the algorithm calculates the following equality:

$$\sum_{j=1}^{(G:K)} \chi_{i_j} \stackrel{?}{=} S, \text{ where } S = \sum_{i=1}^{G:K} i = \frac{(G:K)((G:K)+1)}{2}.$$

This can be done performing the following rotations on $|\psi''_{target}\rangle$ in each of the t subspaces while reading j^{th} bit of the binary input σ:

$$R''_{i,j} = R_{\hat{y}} \left(\frac{2\pi k_i 2^r}{2^n} \right),$$

where the initial state of $|\psi''_{target}\rangle$ was created by $\tilde{R}_i = R_{\hat{y}}(-\frac{2\pi k_i S}{2^n})$ with $S = \frac{G:K(G:K+1)}{2}$.

Assume the set of parameters $K = \{k_1, \ldots, k_t\}$ is "good" according to the Definition 4 and $t = \lceil (2/\epsilon) \ln 2 \cdot 2^n \rceil = O(n)$.

Now consider the circuit representation. Initially $|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_{\log t}\rangle \otimes \left|\psi'_{target}\right\rangle \otimes \left|\psi''_{target}\right\rangle = |00\ldots0\rangle$. For $i \in \{1, \ldots, t\}$, $j \in \{1, \ldots, n\}$ operators $R_{i,j} = (R'_{i,j} \otimes R''_{i,j})$ are combined rotations of $\left|\psi'_{target}\right\rangle$ and $\left|\psi''_{target}\right\rangle$. We define them as follows

$$R'_{i,j} \quad = \quad R_{\hat{y}}\left(\tfrac{2\pi k_i 2^{|K|al}(2^{ql+r} - 2^{(q+1)l+r \bmod |K|l})}{2^n}\right)$$

and

$$R''_{i,j} = \begin{cases} R_{\hat{y}}\left(\frac{2\pi k_i 2^r}{2^n}\right), & \text{when the } j^{th} \text{ input symbol corresponds to the } 1^{st} \\ & \text{ encounter of the coset } a \text{ representative;} \\ I, & \text{otherwise.} \end{cases}$$

The state of the system after having read the input σ is

$$\begin{aligned} |\psi\rangle &= \tfrac{1}{\sqrt{t}} \sum_{i=1}^{t} |i\rangle \left|\psi'_{target}\right\rangle \left|\psi''_{target}\right\rangle; \\ \left|\psi'_{target}\right\rangle &= \cos \tfrac{\pi k_i g_1(\sigma)}{2^n} |0\rangle + \sin \tfrac{\pi k_i g_1(\sigma)}{2^n} |1\rangle; \\ \left|\psi''_{target}\right\rangle &= \cos \tfrac{\pi k_i g_2(\sigma)}{2^n} |0\rangle + \sin \tfrac{\pi k_i g_2(\sigma)}{2^n} |1\rangle; \end{aligned}$$

where

1. $g_1(\sigma) = \sum_a \sum_q 2^{(|K|a+q)\lceil \log G:K \rceil}(\chi_{a,q} - \chi_{a,q-1 \bmod |K|})$. Thus, $g_1(\sigma) = 0$ iff for every coset a function f maps all the elements of a onto the same element of X.

2. $g_2(\sigma) = \left(\sum_{j=1}^{(G:K)} \chi_{i_j}\right) - S$, where χ_{i_j} is the representative chosen from the jth coset. Therefore, $g_2(\sigma)$ checks whether the images of elements from different cosets are distinct.

We accept the input σ if the measurement outcome of $\left|\psi'_{target}\right\rangle \otimes \left|\psi''_{target}\right\rangle$ is $|00\rangle$. Clearly, the accepting probability is

$$Pr_{accept}(\sigma) = \frac{1}{t} \sum_{i=1}^{t} \cos^2 \frac{\pi k_i g_1(\sigma)}{2^n} \cos^2 \frac{\pi k_i g_2(\sigma)}{2^n}$$

When the function f "hides" the subgroup K the acceptance probability is 1. Otherwise, at least one $g_j(\sigma)$ of $g_1(\sigma)$, $g_2(\sigma)$ is not zero and thus $Pr_{accept}(\sigma)$ is bounded as follows

$$Pr_{accept}(\sigma) \leq \frac{1}{t} \sum_{i=1}^{t} \cos^2 \frac{\pi k_i g_j(\sigma)}{2^n} = \frac{1}{2} + \frac{1}{2t} \sum_{i=1}^{t} \cos \frac{2\pi k_i g_j(\sigma)}{2^n} < \frac{1}{2} + \frac{1}{2}\sqrt{\epsilon}$$

since the set $K = \{k_1, \ldots, k_t\}$ is "good".

It is easy to see that the width of the program is linear, while the number of qubits used by our algorithm is $O(\log n)$.

6 Conclusions

Sauerhoff and Sieling in 2004 [SS04] have shown the incomparability between classical and quantum OBDD. Therefore, we consider quantum OBDD complexity of certain important functions.

Using the modified fingerprinting technique we have shown a refined proof of the upper bound for *hidden subgroup problem* [KH05], [Høy97], [ME99] for certain assumptions. The circuit presentation of the results is significantly more illustrative, simplifying the presentation of proofs.

Acknowledgements We thank Juhani Karhumaki for invitation to the University of Turku and a number of interesting discussions on the subject of the paper.

Research was supported by the University of Turku and the Russian Fund for Basic Research (under the grant 08-07-00449).

References

[AK97] F. Ablayev and M. Karpinski, *On the power of randomized ordered branching programs*, Tech. Report 85181-CS, University of Bonn, 1997, see also Electronic Colloquium on Computational Complexity, TR98-004, (1998), available at http://www.eccc.uni-trier.de/eccc/.

[AKK] F. Ablayev, M. Karpinski, and A. Khasianov, *Complexity of Computing Functions on Quantum Branching Programs*, Manuscript, 2008.

[AF98] A. Ambainis and R. Freivalds, *1-way quantum finite automata: strengths, weaknesses and generalization*, Proceeding of the 39th IEEE Conference on Foundation of Computer Science, 1998, See also arXiv:quant-ph/9802062 v3, pp. 332–342.

[AGK01] F. Ablayev, A. Gainutdinova, and M. Karpinski, *On computational power of quantum branching programs*, Lecture Notes in Computer Science, no. 2138, Springer-Verlag, 2001, See also arXiv:quant-ph/0302022 v1, pp. 59–70.

[AGKMP] F. Ablayev, A. Gainutdinova, M. Karpinski, C. Moore, and C. Pollette, *On the computational power of probabilistic and quantum branching programs of constant width*, Information and Computation (2005).

[AN08] A. Ambainis and N. Nahimovs, *Improved constructions of quantum automata*. Manuscript, 2008, from personal communication.

[AV08] F. Ablayev and A. Vasiliev *On the Computation of Boolean Functions by Quantum Branching Programs via Fingerprinting*, TR08-059, (2008), available at http://www.eccc.uni-trier.de/eccc/.

[Fre79] R. Freivalds, *Fast probabilistic algorithms*, FCT'79, LNCS 74 (Berlin, New York), Springer-Verlag, 1979, pp. 57–69.

[Høy97] Peter Høyer, *Conjugated operators in quantum algorithms*, Tech. report, University of Southern Denmar, 1997.

[KH05] A. Khasianov, *Complexity Bounds On Some Fundamental Computational Problems For Quantum Branching Programs*, http://nbn-resolving.de/urn:nbn:de:hbz:5N-05696.

[KH06] A. Khasianov, *Complexity Bounds On Some Fundamental Computational Problems For Quantum Branching Programs*, Manuscript (in Russian), 2006.

[MC97] C. Moore and J.P. Crutchfield, *Quantum automata and quantum grammars*. Theoretical Computer Science 237: 275–306, 2000.

[ME99] M. Mosca and A. Ekert, *The hidden subgroup problem and eigenvalue estimation on a quantum computer*, arXive e-print quant-ph/9903071, 1999.

[MR95] R. Motwani, P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[Nag51] T. Nagell, *Introduction to number theory*, New York: Wiley, 1951.

[NC00] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[NHK00] Masaki Nakanishi, Kiyoharu Hamaguchi, and Toshinobu Kashiwabara, *Ordered quantum branching programs are more powerful than ordered probabilistic branching programs under a bounded-width restriction*, Computing and Combinatorics, LNCS 1858 (Sydney, Australia), 6th Annual International Conference, COCOON 2000, Springer-Verlag, July 2000, pp. 467–476.

[SS04] M. Sauerhoff and Detlef Sieling, *Quantum branching programs and space-bounded nonuniform quantum complexity*, Theoretical Computer Science (2004), no. 334, 177–225.

[Weg00] I. Wegener, *Branching programs and binary decision diagrams*, SIAM Monographs on Discrete Mathematics and Applications, SIAM Press, 2000.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de