Quantum Algorithms for Algebraic Problems *

Sebastian Dörn
Institut für Theoretische Informatik
Universität Ulm
89069 Ulm, Germany

Thomas Thierauf
Fak. Elektronik und Informatik
HTW Aalen
73430 Aalen, Germany

{sebastian.doern,thomas.thierauf}@uni-ulm.de

Abstract

In this paper we present quantum query and time complexity bounds for several group testing problems. For a set S and a binary operation on S, we consider the decision problems whether a given structure with the promise of being a groupoid, semigroup, monoid or quasigroup is in fact a semigroup, monoid, quasigroup or a group. In particular, we give the first application of the new quantum random walk technique by Magniez, Nayak, Roland, and Santha [MNRS07] that improves the previous bounds by Ambainis [Amb04] and Szegedy [Sze04]. Our quantum algorithms for these problems improve the best known classical complexity bounds. We also present upper and lower bounds for testing distributivity and commutativity.

1 Introduction

Quantum algorithms have the potential to demonstrate that for some problems quantum computation is more efficient than classical computation. A goal of quantum computing is to determine for which problems quantum computers are faster than classical computers.

The most important known basic quantum algorithms are Shor's [Sho94] polynomial time factorization algorithm and Grover's search algorithm [Gro96]. Since then, we have seen some generalisations and applications of these two basic quantum techniques. The Shor algorithm has been generalized to quantum algorithms for the hidden subgroup problem (see e.g. [CEMM98]). Grover's search algorithm can be used for quantum amplitude amplification [BHMT02] and quantum random walk search [Amb04, Sze04, MNRS07]. The application of these quantum search tools is a fast growing area in quantum computing. For example, quantum algorithms have been presented for several problems from computer science (see e.g. [BHT98, BDHHMSW01, Amb04]), graph theory (see e.g. [DHHM04, MSS05, AS06, Doe07a, Doe07b]) and (linear) algebra (see e.g. [MN05, BS06, DT07, DT08a, DT08b]).

*Supported by DFG grants Scho 302/7-2.

In this paper we study the quantum complexity of group testing problems. For a set S of size n and a binary operation on S, we consider the decision problems whether a given structure with the promise of being a groupoid, semigroup, monoid or quasigroup is in fact a semigroup, monoid, quasigroup or a group. In particular, we give the first application of the new quantum random walk technique by Magniez, Nayak, Roland, and Santha [MNRS07] that improves the previous bounds by Ambainis [Amb04] and Szegedy [Sze04]. We present also upper and lower bounds for testing distributivity and commutativity. This paper is a summary of our conference papers from FCT'07 [DT07] and SOFSEM'08 [DT08b].

The motivation for studying the query complexity of algebraic problems is twofold. On the one hand side, these are fundamental and basic problems which have many applications in computer science. For example, testing if a black box is a group is very useful in cryptography. On the other hand, we can analyze how powerful are our tools for the construction of lower and upper bounds for the quantum query complexity of these problem. For many problems we can find optimal quantum algorithms by a combination of Grover search, amplitude amplification and quantum walk search. But for some problems this doesn't seem to work. Maybe this can be a motivation for the development of new quantum techniques.

In this paper our input is an operation table for a set S of size $n \times n$. In Section 3 we consider the semigroup problem, that is, whether the operation on S is associative. Rajagopalan and Schulman [RS00] developed a randomized algorithm for this problem that runs in time $O(n^2)$. As an additional parameter, we consider the binary operation $\circ : S \times S \to S'$, where $S' \subseteq S$. We construct a quantum algorithm for this problem whose query complexity is $O(n^{5/4})$, if the size of S' is constant. Our algorithm is the first application of the new quantum random walk search scheme by Magniez, Nayak, Roland, and Santha [MNRS07]. With the quantum random walk of Ambainis [Amb04] and Szegedy [Sze04], the query complexity of our algorithm would not improve the obvious Grover search algorithm for this problem. Furthermore we show a quantum query lower bound for the semigroup problem of $\Omega(n)$, which holds also if the size of S' is constant.

In Section 4 we have given a finite set S of size n with a binary operation $\circ : S \times S \to S$ represented by a table. One has to decide whether S has an identity element or is a monoid. We show that the identity problem can be solved with linearly many quantum queries. This is optimal, since we also prove a tight lower bound for this problem. Moreover we show a linear lower bound of the quantum query complexity for testing whether a groupoid is a monoid.

In Section 5 we consider several group problems. Given a groupoid, semigroup, monoid or quasigroup S by its operation table, we have to decide whether S is a group. We present a randomized algorithm for testing whether a semigroup resp. monoid is a group with running time of $O(n^{\frac{3}{2}})$. This improves the naive $O(n^2)$ algorithm that searches for an inverse in the operation table for every element. Then we show that on a quantum computer the complexity can be improved to $\widetilde{O}(n^{\frac{11}{14}})$. Furthermore we give nearly optimal quantum query algorithms for testing whether a groupoid or quasigroup is a group.

In Section 6 we present several bounds for testing commutativity. We prove that the quantum query complexity of the commutativity problem for groupoids, semigroups and monoids is $\Theta(n)$. In addition, we show that the commutativity problem can be solved in logarithmic number of quantum queries to the operation table if it is a quasigroup resp. group.
In Section 7 we consider the distributive problem, given a set S and two binary operations $\oplus : S \times S \to S$ and $\otimes : S \times S \to S$ represented by a table. One has to decide whether $(S, \oplus, \otimes)$ is distributive, i.e. we have to test whether the two equations $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ are satisfied. We show a linear lower bound on the quantum query complexity for this problem. Moreover we prove that the distributive problem can be decided with linear quantum query complexity, if $(S, \oplus)$ is a commutative quasigroup.

2 Preliminaries

2.1 Algebraic Structures

Let S be some set and $\circ$ a binary operation on S. Then the pair $(S, \circ)$ is called a *groupoid*. If $\circ$ is associative, then $(S, \circ)$ is a *semigroup*. If there is an identity element in addition, then $(S, \circ)$ is a *monoid*. If, moreover, every element has an inverse, then $(S, \circ)$ is a *group*.

The operation $\circ$ is *cancellative*, if for every $a, b \in S$ the equations $a \circ x = b$ and $x \circ a = b$ have a unique solution, respectively. When $\circ$ is cancellative, $(S, \circ)$ is called a *quasigroup*. Clearly, every group is a quasigroup. We mostly assume that $(S, \circ)$ is given by an operation table T. Then $(S, \circ)$ is a quasigroup iff every row and column of T is a permutation of S. A quasigroup $(S, \circ)$ is a *loop*, if it has an identity element.

2.2 Quantum Query Model

In the query model, the input $x_1, \ldots, x_N$ is contained in a black box or oracle and can be accessed by queries to the black box. As a query we give i as input to the black box and the black box outputs x_i. The goal is to compute a Boolean function $f : \{0,1\}^N \to \{0,1\}$ on the input bits $x = (x_1, \ldots, x_N)$ minimizing the number of queries. The classical version of this model is known as decision tree.

The quantum query model was explicitly introduced by Beals et al. [BBCMW01]. In this model we pay for accessing the oracle, but unlike the classical case, we use the power of quantum parallelism to make queries in superposition. The state of the computation is represented by $|i, b, z\rangle$, where i is the query register, b is the answer register, and z is the working register. A quantum computation with T queries is a sequence of unitary transformations

$$U_0 \to O_x \to U_1 \to O_x \to \ldots \to U_{T-1} \to O_x \to U_T,$$

where each U_j is a unitary transformation that does not depend on the input x, and O_x are query (oracle) transformations. The oracle transformation O_x can

be defined as $O_x : |i, b, z\rangle \rightarrow |i, b \oplus x_i, z\rangle$. The computations consists of the following three steps:

1. Go into the initial state $|0\rangle$.
2. Apply the transformation $U_T O_x \cdots O_x U_0$.
3. Measure the final state.

The result of the computation is the rightmost bit of the state obtained by the measurement.

The quantum computation determines f with bounded error, if for every x, the probability that the result of the computation equals $f(x_1, \ldots, x_N)$ is at least $1-\epsilon$, for some fixed $\epsilon < 1/2$. In the query model of computation each query adds one to the query complexity of an algorithm, but all other computations are free. The time complexity of the algorithm is usually measured in terms of the total circuit size for the unitary operations U_i. All quantum algorithms in this paper are bounded error.

The quantum query complexity of black box computation has become a great interest in quantum computing. The black box model provides a simple and abstract framework for the construction of quantum algorithms. All quantum algorithms can be formulated in the black box model, we can determine the speed up against classical algorithm, and we can prove lower bounds for the quantum query complexity.

2.3 Tools for Quantum Algorithms

For the basic notation on quantum computing, we refer the reader to the textbook by Nielsen and Chuang [NC03]. Here, we give three tools for the construction of our quantum algorithms.

2.3.1 Quantum Search.

A search problem is a subset $P \subseteq \{1, \ldots, N\}$ of the search space $\{1, \ldots, N\}$. With P we associate its characteristic function $f_P : \{1, \ldots, N\} \rightarrow \{0, 1\}$ with

$$f_P(x) = \begin{cases} 1, & \text{if } x \in P, \\ 0, & \text{otherwise.} \end{cases}$$

Any $x \in P$ is called a solution to the search problem. Let $k = |P|$ be the number of solutions of P.

Theorem 2.1 [Gro96, BBHT98, BCWZ99] *Let $P \subseteq [N]$ be a search problem and k the number of solutions of P.*

1. *Finding one solution of P can be done in $O(\sqrt{N/k})$ expected quantum queries to f_P with probability of at least a constant. The search algorithm does not require prior knowledge of k.*

2. *Finding one solution of P can be done in $O(\sqrt{N})$ quantum queries to f_P with probability of at least a constant, provided there is one.*

3. *Whether $k > 0$ can be decided in $O(\sqrt{N})$ quantum queries to f_P with probability of at least a constant.*

4. *Finding all solutions of P can be done in $O(\sqrt{kN})$ quantum queries to f_P with probability of at least a constant.*

The running time complexity of Grover search is larger than its query complexity by a logarithmic factor.

2.3.2 Amplitude Amplification.

The quantum amplitude amplification is a generalization of Grover's search algorithm [Gro96]. Let $\mathcal{A}$ be an algorithm for a problem with small success probability at least ϵ. Classically, we need $\Theta(1/\epsilon)$ repetitions of $\mathcal{A}$ to increase its success probability from ϵ to a constant, for example 2/3. The corresponding technique in the quantum case is called amplitude amplification.

Theorem 2.2 [BHMT02] *Let $\mathcal{A}$ be a quantum algorithm with one-sided error and success probability at least ϵ. Then there is a quantum algorithm $\mathcal{B}$ that solves $\mathcal{A}$ with success probability 2/3 by $O(\frac{1}{\sqrt{\epsilon}})$ invocations of $\mathcal{A}$.*

2.3.3 Quantum Walk.

Quantum walks are the quantum counterpart of Markov chains and random walks. The quantum walk search provide a promising source for new quantum algorithms, like element distinctness algorithm [Amb04], triangle finding [MSS05], testing group commutativity [MN05], matrix verification [BS06]. Let $P = (p_{xy})$ be the transition matrix of an ergodic symmetric Markov chain on the state space X.

Let $M \subseteq X$ be a set of marked states. Assume that the search algorithms use a data structure D that associates some data $D(x)$ with every state $x \in X$. From $D(x)$, we would like to determine if $x \in M$. When operating on D, we consider the following three types of costs:

- *Setup cost s*: The worst case cost to compute $D(x)$, for $x \in X$.
- *Update cost u*: The worst case cost for transition from x to y, and update $D(x)$ to $D(y)$.
- *Checking cost c*: The worst case cost for checking if $x \in M$ by using $D(x)$.

Magniez et al. [MNRS07] developed a new scheme for quantum search, based on any ergodic Markov chain. Their work generalizes previous results by Ambainis [Amb04] and Szegedy [Sze04]. They extend the class of possible Markov chains and improve the query complexity as follows.

Theorem 2.3 [MNRS07] *Let $\delta > 0$ be the eigenvalue gap of an ergodic Markov chain P and let $\frac{|M|}{|X|} \geq \epsilon$. Then there is a quantum algorithm that determines if M is empty or finds an element of M with cost*

$$s + \frac{1}{\sqrt{\epsilon}} \left(\frac{1}{\sqrt{\delta}} u + c \right).$$

In the most practical applications (see [Amb04, MSS05]) the quantum walk takes place on the Johnson graph $J(n, r)$, which is defined as follows: the vertices are subsets of $\{1, \ldots, n\}$ of size r and two vertices are connected iff they differ in exactly one number. It is well known, that the spectral gap δ of $J(n, r)$ is $\Theta(1/r)$ for $1 \leq r \leq \frac{n}{2}$.

We apply the quantum walk on the graph categorical product of two Johnson graphs. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs, the *graph categorical product* $G = (V, E) = G_1 \times G_2$ of G_1, G_2 is defined as follows: $V = V_1 \times V_2$, and $((g_1, g_2), (g_1', g_2')) \in E$ iff $(g_1, g_1') \in E_1$ and $(g_2, g_2') \in E_2$.

2.4 Tool for Quantum Query Lower Bounds

In this paper, we use the following special case of a method by Ambainis [Amb02] to prove lower bounds for the quantum query complexity.

Theorem 2.4 [Amb02] *Let $\mathcal{F} = \{f : [n] \times [n] \to [n]\}$ be the set of all possible input function, and $\Phi : \mathcal{F} \to \{0, 1\}$. Let $A, B \subset \mathcal{F}$ such that $\Phi(f) = 1$ and $\Phi(g) = 0$ for all $f \in A$ and $g \in B$. Let $R \subset A \times B$, and m, m', l, l' be numbers such that*

1. *for every $f \in A$, there are at least m different $g \in B$ such that $(f, g) \in R$.*
2. *for every $g \in B$, there are at least m' different $f \in A$ such that $(f, g) \in R$.*
3. *for every $f \in A$ and $x, y \in [n]$, there are at most l different $g \in B$ such that $(f, g) \in R$ and $f(x, y) \neq g(x, y)$.*
4. *for every $g \in B$ and $x, y \in [n]$, there are at most l' different $f \in A$ such that $(f, g) \in R$ and $f(x, y) \neq g(x, y)$.*

Then every bounded-error quantum algorithm that computes Φ has quantum query complexity $\Omega\left(\sqrt{\frac{m \cdot m'}{l \cdot l'}}\right)$.

Let $f, g : [n] \times [n] \to [n]$ be two functions, we define

$$d(f, g) = |\{\, x, y \in [n] \mid f(x, y) \neq g(x, y) \,\}|.$$

In some cases, we consider the special case $A, B \subset \{0, 1\}^{n \times n}$ and $(f, g) \in R$ if and only if f and g differ in exactly one position. Then it is $l = l' = 1$, and every bounded-error quantum algorithm that computes f has quantum query complexity of $\Omega\left(\sqrt{m \cdot m'}\right)$.

3 The Semigroup Problem

In the semigroup problem we are given two sets S and $S' \subseteq S$ and a binary operation $\circ : S \times S \to S'$ represented by a table. We denote with n the size of the set S. One has to decide whether S is a semigroup, that is, whether the operation on S is associative.

The complexity of this problem was first considered by Rajagopalan and Schulman [RS00], who gave a randomized algorithm with time complexity $O(n^2 \log \frac{1}{\delta})$, where δ is the error probability. They also showed a lower bound of $\Omega(n^2)$. The previously best known algorithm was the naive $O(n^3)$-algorithm that checks all triples.

In the quantum setting, one can do a Grover search over all triples $(a, b, c) \in S^3$ and check whether they are associative. The quantum query complexity of the search is $O(n^{3/2})$. We construct a quantum algorithm for the semigroup problem that has query complexity $O(n^{5/4})$, if the size of S' is constant. Furthermore we give a quantum query lower bound of $\Omega(n)$ for this problem. Our algorithm is the first application of the recent quantum random walk search scheme by Magniez et al. [MNRS07]. The quantum random walk of Ambainis [Amb04] and Szegedy [Sze04] doesn't suffice to get an improvement of the Grover search mentioned above.

Theorem 3.1 *Let $k = n^\alpha$ be the size of S' with $0 < \alpha \leq 1$. The quantum query complexity of the semigroup problem is*

$$\begin{cases} O(n^{\frac{5+\alpha}{4}}), & \text{for } 0 \leq \alpha \leq \frac{1}{3}, \\ O(n^{\frac{6+2\alpha}{5}}), & \text{for } \frac{1}{3} < \alpha \leq \frac{3}{4}, \\ O(n^{\frac{3}{2}}), & \text{for } \frac{3}{4} < \alpha \leq 1. \end{cases}$$

Proof. We use the quantum walk search scheme of Theorem 2.3. To do so, we construct a Markov chain and a database for checking if a vertex of the chain is marked.

Our quantum walk is done on the categorical graph product G_J of two Johnson graphs $J(n, r)$. Let A and B be two subsets of S of size r. We will determine r later. We search for a pair $(a, b) \in S^2$, such that a, b are two elements of a non-associative triple. Then the marked vertices of G_J correspond to pairs (A, B) with $(A \circ B) \circ S \neq A \circ (B \circ S)$. In every step of the walk, we exchange one row and one column of A and B.

The database of our quantum walk is the set

$$D(A, B) = \{\, (a, b, a \circ b) \mid a \in A \cup S' \text{ and } b \in B \cup S' \,\}.$$

Now we compute the quantum query costs for the setup, update and checking. The setup cost for the database $D(A, B)$ is $O((r + k)^2)$ and the update cost is $O(r + k)$.

To check whether a pair (A, B) is marked, we have to test if $(A \circ B) \circ S \neq A \circ (B \circ S)$. We claim that the quantum query cost to check this inequality is $O(\sqrt{nrk})$: fix a pair $(b, c) \in B \times S$. We check whether $(A \circ b) \circ c \neq A \circ (b \circ c)$. To

get $(b \circ c)$ we make one query to the oracle. Because $(b \circ c) \in S'$, the computation of $A \circ (b \circ c)$ can be done by using our database. We obtain r values which we denote by $(y_1, \ldots, y_r)$. The evaluation of $(A \circ b)$ needs no queries by using our database, let $(z_1, \ldots, z_r)$ be the result. Note that $z_i \in S'$ for all i. Now we use Grover's algorithm for searching a $s \in S'$ such that $s \circ c \neq y_j$ for a $j \in [r]$ with $s = z_j$. This search can be done in $O(\sqrt{k})$ quantum queries. The outer loop is a Grover search for a pair $(b, c) \in B \times S$. Therefore, the total checking cost is $O(\sqrt{nrk})$.

The spectral gap of the walk on G_J is $\delta = O(1/r)$ for $1 \leq r \leq \frac{n}{2}$, see [BS06]. If there is a triple (a, b, c) with $(a \circ b) \circ c \neq a \circ (b \circ c)$, then there are at least $\binom{n-1}{r-1}^2$ marked sets (A, B). Therefore we have

$$\varepsilon \geq \frac{|M|}{|X|} \geq \left(\frac{\binom{n-1}{r-1}}{\binom{n}{r}} \right)^2 = \frac{r^2}{n^2}.$$

Let $r = n^\beta$ for $0 < \beta < 1$. Assuming $r > k$, then the quantum query complexity of the semigroup problem is

$$O\left(r^2 + \frac{n}{r}\left(\sqrt{r} \cdot r + \sqrt{nrk}\right)\right) = O\left(n^{2\beta} + n^{1+\frac{\beta}{2}} + n^{\frac{3+\alpha-\beta}{2}}\right).$$

Now we choose β depending on α such that this expression is minimal. Suppose that $2\beta \leq 1 + \frac{\beta}{2}$, i.e. $\beta \leq \frac{2}{3}$. From the equation $1 + \frac{\beta}{2} = \frac{3+\alpha-\beta}{2}$, we get $\beta = \frac{1+\alpha}{2}$. Then the quantum query complexity of the semigroup problem is $O(n^{\frac{5+\alpha}{4}})$ for $r = n^{\frac{1+\alpha}{2}}$ and $\alpha \leq \frac{1}{3}$. Otherwise if $2\beta > 1 + \frac{\beta}{2}$, i.e. $\beta > \frac{2}{3}$, we get $\beta = \frac{3+\alpha}{5}$ from the equation $2\beta = \frac{3+\alpha-\beta}{2}$. Then the quantum query complexity is $O(n^{\frac{6+2\alpha}{5}})$ for $r = n^{\frac{3+\alpha}{5}}$ and $\alpha > \frac{1}{3}$. If $\alpha > \frac{3}{4}$, the query complexity is bigger than $O(n^{\frac{3}{2}})$, therefore we use Grover search instead of quantum walk search. □

For the special case that $\alpha = 0$, i.e., only a constant number of elements occur in the operation table, we get

Corollary 3.2 *The quantum query complexity of the semigroup problem is $O(n^{\frac{5}{4}})$, if S' has constant size.*

Note that the time complexity of our algorithm is $O(n^{1.5} \log n)$.

Theorem 3.3 *The semigroup problem requires $\Omega(n)$ quantum queries.*

Proof. Let S be a set of size n and $\circ : S \times S \to \{0, 1\}$ a binary operation represented by a table. We apply Theorem 2.4. The set A consists of all $n \times n$ matrices, where the entry of position $(1, 1)$, $(1, c)$, $(c, 1)$ and (c, c) is 1, for $c \in S - \{0, 1\}$, and zero otherwise. The operation tables of A are associative, since $(x \circ y) \circ z = x \circ (y \circ z) = 1$ for all $x, y, z \in \{1, c\}$ and zero otherwise.

The set B consists of all $n \times n$ matrices, where the entry of position $(1, 1)$, $(1, c)$, $(c, 1)$, (c, c) and (a, b) is 1, for fixed $a, b, c \in S - \{0, 1\}$ with $a, b \neq c$, and zero otherwise. Then $(a \circ b) \circ c = 1$ and $a \circ (b \circ c) = 0$. Therefore the operation tables of B are not associative.

From each $T \in A$, we can obtain $T' \in B$ by replacing the entry 0 of T at (a, b) by 1, for any $a, b \notin \{0, 1, c\}$. Hence we have $m = \Omega(n^2)$. From each $T' \in B$, we can obtain $T \in A$ by replacing the entry 1 of T' at position (a, b) by 0, for $a, b \notin \{0, 1, c\}$. Then we have $m' = 1$. By Theorem 2.4, the quantum query complexity is $\Omega(\sqrt{m \cdot m'}) = \Omega(n)$. □

From our proof follows, that the lower bound holds also for constant size of S'.

4 The Monoid Problem

In the monoid problem we are given a finite set S of size n with a binary operation $\circ : S \times S \to S$ represented by a table. One has to decide whether S is a monoid.

The monoid problem is an extention of the semigroup problem of the previous section. We have to verify whether the groupoid $(S, \circ)$ is associative and has an identity element. We show that the identity problem requires linearly many quantum queries. We start by considering the 1-column problem: given a 0-1-matrix of order n, decide whether it contains a column that is all 1.

Lemma 4.1 *The 1-column problem requires $\Omega(n)$ quantum queries.*

Proof. We use Theorem 2.4. The set A consists of all matrices, where in $n-1$ columns there is exactly one entry with value 0, and the other entries of the matrix are 1. The set B consists of all matrices, where in every column there is exactly one entry with value 0, and the other entries of the matrix are 1. From each matrix $T \in A$, we can obtain $T' \in B$ by changing one entry in the 1-column from 1 to 0. Then we have $m = n$. From each matrix $T' \in B$, we can obtain $T \in A$ by changing one entry from 0 to 1. Then we have $m' = n$. By Theorem 2.4, the quantum query complexity is $\Omega(n)$. □

Theorem 4.2 *The identity problem requires $\Omega(n)$ quantum queries.*

Proof. We reduce the 1-column problem to the identity problem. Given a 0-1-matrix $M = (m_{i,j})$ of order n. We define $S = \{0, 1, \ldots, n\}$ and a operation table $T = (t_{i,j})$ with $0 \leq i, j \leq n$ for S as follows:

$$t_{i,j} = \begin{cases} 0, & \text{if } m_{i,j} = 0, \\ i, & \text{if } m_{i,j} = 1, \end{cases}$$

and $t_{0,j} = t_{i,0} = 0$. Then M has a 1-column iff T has an identity element. □

Finding an identity element is simple. We choose an element $a \in S$ and then we test if a is the identity element by using Grover search in $O(\sqrt{n})$ quantum queries. The success probability of this procedure is $\frac{1}{n}$. By using the amplitude amplification we get an $O(n)$ quantum query algorithm for finding an identity element (if there is one). Since the upper and the lower bound match, we have determined the precise complexity of the identity problem.

Corollary 4.3 *The quantum query complexity of the identity problem is* $\Theta(n)$.

Corollary 4.4 *Whether a groupoid is a monoid requires* $\Omega(n)$ *quantum queries.*

5 Group Problems

In this section we consider the decision problems whether a given structure with the promise of being a groupoid, semigroup, monoid or quasigroup S of size n with a binary operation $\circ$ is in fact a group.

5.1 Group testing for Monoids

We consider the problem whether a given finite monoid M is in fact a group. That is, we have to check whether every element of M has an inverse. The monoid M has n elements and is given by its operation table and the identity element e.

To the best of our knowledge, this special group problem has not been studied before. The naive approach for the problem checks for every element $a \in M$, whether e occurs in a's row in the operation table. The query complexity is $O(n^2)$. We develop a (classical) randomized algorithm that solves the problem with $O(n^{\frac{3}{2}})$ queries to the operation table. Then we show that on a quantum computer the query complexity can be improved to $\widetilde{O}(n^{\frac{11}{14}})$.

Theorem 5.1 *Whether a given monoid is a group can be decided with*

1. $O(n^{\frac{3}{2}})$ *queries by a randomized algorithm.*
2. $O(n^{\frac{11}{14}} \log n)$ *by a quantum query algorithm.*

Proof. We start by presenting the classical algorithm. Let $a \in M$, we consider the sequence of powers $a, a^2, a^3, \ldots$. Since M is finite, there will be a repetition at some point. We define the *order of* a as the smallest power t, such that $a^t = a^s$, for some $s < t$. Clearly, if a has an inverse, s must be zero.

Lemma 5.2 *Let* $a \in M$ *of order* t. *Then* a *has an inverse iff* $a^t = e$.

Hence the powers of a will tell us at some point whether a has an inverse. On the other hand, if a has no inverse, the powers of a provide more elements with no inverse as well.

Lemma 5.3 *Let* $a \in M$. *If* a *has no inverse, then* a^k *has no inverse, for all* $k \geq 1$.

Our algorithm has two phases. In phase 1, it computes the powers of every element up to certain number r. That is, we consider the sequences $S_r(a) = (a, a^2, \ldots, a^r)$, for all $a \in M$. If $e \in S_r(a)$ then a has an inverse by Lemma 5.2. Otherwise, if we find a repetition in the sequence $S_r(a)$, then, again by Lemma 5.2, a has no inverse and we are done.

If we are not already done by phase 1, i.e. there are some sequences $S_r(a)$ left such that $e \notin S_r(a)$ and $S_r(a)$ has pairwise different elements. Then the algorithm proceeds to phase 2. It selects some $a \in M$ uniformly at random and checks whether a has an inverse by searching for e in the row of a in the operation table. This step is repeated n/r times.

For the correctness observe that the algorithm accepts with probability 1 if M is a group. Now assume that M is not a group. Assume further that the algorithm does not already detect this in phase 1. Let a be some element without an inverse. By Lemma 5.2, the sequence $S_r(a)$ has r pairwise different elements which don't have inverses too by Lemma 5.3. Therefore in phase 2, the algorithm picks an element without an inverse with probability of at least r/n. By standard arguments, the probability that at least one out of n/r many randomly chosen elements has no inverse, is constant.

The query complexity of the algorithm is bounded by rn in phase 1 and by n^2/r in phase 2. Total the query complexity of the algorithm is

$$O\left(nr + n^2/r\right),$$

which is minimized for $r = n^{\frac{1}{2}}$. Hence the query complexity for testing if a semigroup is a group, is $O(n^{\frac{3}{2}})$.

For the quantum query complexity we use Grover search and amplitude amplification. In phase 1, we search for an $a \in M$, such that the sequence $S_r(a)$ has r pairwise different entries different from e. This property can be checked by first searching $S_r(a)$ for an occurance of e by a Grover search with $\sqrt{r}\log r$ queries. Then, if e doesn't occur in $S_r(a)$, we check whether there is an element in $S_r(a)$ that occurs more than once. This is the element distinctness problem and can be solved with $r^{2/3}\log r$ queries, see [Amb04]. Therefore the quantum query complexity of phase 1 is bounded by $\sqrt{n}\cdot r^{2/3}\log r$. In phase 2 we search for an $a \in M$ such that a has no inverse. Therefore we actually search the row of a in the operation table. Hence this takes $\sqrt{n}$ queries. Since at least r of the a's don't have an inverse, by amplitude amplification we get $\sqrt{n}\sqrt{n/r} = n/\sqrt{r}$ queries in phase 2. In summary, the quantum query complexity is

$$O(\sqrt{n}\cdot r^{2/3}\log r + \frac{n}{\sqrt{r}}),$$

which is minimized for $r = n^{\frac{3}{7}}$. Hence we have a $O(n^{\frac{11}{14}}\log n)$ quantum query algorithm. □

The time complexity of our classical algorithm is $O(n^{\frac{3}{2}})$. Our quantum implementation has nearly quadratic speed up over the classical algorithm. In the quantum algorithm we have used several Grover search subroutines, one amplitude amplification, and one application of the quantum walk element distinctness procedure by Ambainis [Amb04]. Therefore the quantum time complexity is $O(n^{\frac{11}{14}}\log^c n)$ for a constant c, since the element distinctness procedure has running time of $O(n^{\frac{2}{3}}\log^c n)$.

Corollary 5.4 *The time complexity of the group testing algorithm is $O(n^{\frac{3}{2}})$ in the classical setting and $O(n^{\frac{11}{14}}\log^c n)$ in the quantum setting.*

5.2 Group testing for Semigroups

Now we consider the problem whether a finite semigroup $(S, \circ)$ is in fact a group. The naive approach for this problem searches first for an identity element e of S and then checks whether e occurs in every row of the operation table. The query complexity of this procedure is $O(n^2)$, resp. $O(n)$ in the quantum case.

Theorem 5.5 *Whether a given semigroup is a group can be decided with*

1. $O(n^{\frac{3}{2}})$ *queries by a randomized algorithm.*
2. $O(n^{\frac{11}{14}} \log n)$ *by a quantum query algorithm.*

Proof. Our input is a finite semigroup $(S, \circ)$, and we want to decide whether it is in fact a group. To do so, we first search for an identity element and then use the algorithm of Theorem 5.1. To find the identity element, we start by choosing an element a of S and search for an element $e \in S$ such that $a \circ e = a$. Then e is our candidate for the identity element. Recall that we finally want to decide whether S is a group. In this case, the identity element is unique. Hence if our candidate e doesn't work we can safely reject the input, even in the case that S actually has an identity element. To test our candidate e, it suffices to check whether $b \circ e = b$ for all $b \in S$. Obviously the two steps can be done in $O(n)$ queries classically and $O(\sqrt{n})$ quantum queries with Grover search. □

The result should be contrasted with the following: if we want to decide whether a given semigroup is in fact a monoid, then the best known algorithms make $O(n^2)$ queries classically and $O(n)$ queries in the quantum setting.

5.3 Group testing for Quasigroups

Next we assume that the input $(S, \circ)$ is a quasigroup. Rajagopalan and Schulman [RS00] showed, that in a quasigroup we can deterministically compute a set of generators of size $\log n$ in quadratic time. Light observed (see [CP61]) that if $R \subset S$ is a set of generators of S, then it suffices to test all triples a, b, c in which b is an element of R. Therefore Light's observation results in an $O(n^2 \log n)$ deterministic algorithm for verifying associativity of quasigroups.

Theorem 5.6 *Whether a given quasigroup or a loop is a group can be decided with expected quantum query complexity of* $\Theta(n)$.

Proof. First we prove the upper bound. We have to verify if the quasigroup $(S, \circ)$ is associative. Therefore we choose three elements $a, b, c \in S$, and then we verify if $(a \circ b) \circ c \neq a \circ (b \circ c)$. Rajagopalan and Schulman [RS00] showed that any non-associative quasigroup has at least $n - 2$ non-associative triples. Therefore the success probability for finding a non-associative triple (if there is one) is at least $\frac{n-2}{n^3}$. By using the quantum amplitude amplification we have an $O(n)$ quantum query algorithm for finding a non-associative triple in a quasigroup (if there is one).

For the lower bound, we apply Theorem 2.4 in connection with an idea of [RS00] for proving an $\Omega(n^2)$ lower bound for this problem in classical computing. The set A consists of the operation table T of the group $(\mathbb{Z}_2^m, +)$, where

$+$ is the vector addition modulo 2. Let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}_2^m$ with $\boldsymbol{a} \neq \boldsymbol{0}$. The set B consists of all operation tables of $(\mathbb{Z}_2^m, \circ)$, where $\circ$ is equal to $+$ except in the following four positions:

1. $\boldsymbol{b} \circ \boldsymbol{c} = \boldsymbol{b} + (\boldsymbol{a} + \boldsymbol{c})$,
2. $\boldsymbol{b} \circ (\boldsymbol{a} + \boldsymbol{c}) = \boldsymbol{b} + \boldsymbol{c}$,
3. $(\boldsymbol{a} + \boldsymbol{b}) \circ \boldsymbol{c} = \boldsymbol{b} + \boldsymbol{c}$,
4. $(\boldsymbol{a} + \boldsymbol{b}) \circ (\boldsymbol{a} + \boldsymbol{c}) = \boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c}$.

All tables of B are quasigroups, because the above modifications simply exchange two elements in two rows of the table T, but they are not associative, since

$$\boldsymbol{a} + \boldsymbol{b} = (\boldsymbol{c} \circ (\boldsymbol{a} + \boldsymbol{b})) \circ \boldsymbol{c} \neq \boldsymbol{c} \circ ((\boldsymbol{a} + \boldsymbol{b}) \circ \boldsymbol{c}) = \boldsymbol{b}.$$

The relation R is defined by

$$R = \{\, (T, T') \in (A, B) \mid T' \text{ originates of the above four modifications of } T \,\}.$$

Then R satisfies $m = \Omega(n^3),\ m' = 1,\ l = \Omega(n)$ and $l' = 1$. □

5.4 Group testing for Groupoids

We consider the problem whether an arbitrary $(S, \circ)$ is in fact a group. There is a $O(n^2 \log n)$ deterministic algorithm for this problem by [RS00]. We develop a quantum algorithm that has time complexity $O(n^{\frac{13}{12}} \log^2 n)$. Furthermore, we present an $O(n \log n)$ query algorithm for this problem, that has a time complexity $O(n^2 \log n)$ however. The latter algorithm is nearly optimal with respect to the query complexity, as we prove a linear lower bound for this problem.

We need a generalization of a lemma from [RS00]. First we generalize the notion of a cancellative operation.

Definition 5.7 *Let $(S, \circ)$ be a groupoid with n elements represented by its operation table T. Let $I, J \subseteq [n]$ be two index sets and let $T_{I,J}$ be the subtable of T indexed by I and J. We call* $\circ$ cancellative on $T_{I,J}$*, if every element occurs at most once in every row and ervery column of $T_{I,J}$.*

Lemma 5.8 [RS00] *Let $\circ$ be cancellative on a $r \times n$ subtable of its operation table. If $\circ$ is non-associative, then it has at least $r/4$ non-associative triples.*

Proof. Let (a, b, c) be a non-associative triple and $a = a' \circ a''$. Consider the following cycle of equations:

$$\begin{aligned}
(a' \circ a'') \circ (b \circ c) &= ((a' \circ a'') \circ b) \circ c) \\
&= (a' \circ (a'' \circ b)) \circ c) \\
&= a' \circ ((a'' \circ b)) \circ c)) \\
&= a' \circ (a'' \circ (b \circ c)) \\
&= (a' \circ a'') \circ (b \circ c).
\end{aligned}$$

Every equation is an application of the associativity law. Since (a, b, c) is a non-associative triple, the first equation fails. Therefore at least one of the other equations must fail as well. Hence at least one of the following four triples must be non-associative:

1. (a', a'', b),
2. $(a', a'' \circ b, c)$,
3. (a'', b, c),
4. $(a', a'', b \circ c)$.

If $\circ$ is cancellative on a $r \times n$ subtable, then a can be written as $a' \circ a''$ in r different ways. Then the associativity fails in at least one of the four categories for each of these r pairs. Hence there is a category for which there are at least $r/4$ failures. Since each category identifies either a' or a'', there are no duplicate triples in any category. □

Theorem 5.9 *Whether a groupoid is a group can be decided by a quantum algorithm within $O(n^{\frac{13}{12}} \log^c n)$ expected steps, for some constant c.*

Proof. Let $(S, \circ)$ be a groupoid represented by its operation table T. Recall that if S is a group, then $\circ$ is cancellative. Our first step is to determine whether the operation is associative. To do so, we choose an arbitrary subset A of S of size r. We determine r later. Then we check whether $\circ$ is cancellative on the subtable T_A of T, where T_A is the $r \times n$ table that consists of the rows of T indexed by A. This is not the case, if we find a row or column in T_A with two equal elements. Hence we can solve this with a Grover search and the element distinctness quantum algorithm by Ambainis [Amb04]. The quantum query complexity of this procedure is $O(\sqrt{r}n^{\frac{2}{3}} + \sqrt{n}r^{\frac{2}{3}})$.

If any of the considered rows and colmuns are not cancellative then we are done. Otherwise we randomly choose three elements $a, b, c \in S$ and check whether $(a \circ b) \circ c \neq a \circ (b \circ c)$. If the operation is not associative, then the probability of finding a non-associative triple is at least $\frac{r}{4n^3}$ by Lemma 5.8. By using the quantum amplitude amplification we have an $O(n^{\frac{3}{2}}/\sqrt{r})$ quantum query algorithm for finding a non-associative triple.

If there is no non-associative triple, then $(S, \circ)$ is a semigroup. Whether this semigroup is a group can be decided with $O(n^{\frac{11}{14}} \log n)$ quantum queries by Theorem 5.5. The total expected quantum query complexity of this algorithm we get

$$O\left(\sqrt{r}n^{\frac{2}{3}} + \sqrt{n}r^{\frac{2}{3}} + \frac{n^{\frac{3}{2}}}{\sqrt{r}} + n^{\frac{11}{14}} \log n\right).$$

This expression is minimized for $r = n^{\frac{5}{6}}$. Hence the expected time complexity of this algorithm is $O(n^{\frac{13}{12}} \log^c n)$ for a constant c. □

By setting $r = n$ in Lemma 5.8, we have

Corollary 5.10 *Whether a groupoid is a quasigroup can be decided by a quantum algorithm within $O(n^{\frac{7}{6}} \log n)$ expected steps.*

We can further improve the query complexity of the problem, if we allow a larger running time.

Theorem 5.11 *Whether a groupoid is a group can be decided with $O(n \log n)$ expected quantum queries.*

Proof. Let $(S, \circ)$ be a groupoid represented by its operation table T. A well known fact from algebra is, that if $(S, \circ)$ is a quasigroup, then a random subset $R \subset S$ with $c \log n$ elements is a set of generators with probability at least $1 - \exp(c)$ (see [RS00]). We choose a random subset R of $O(\log n)$ elements of S. Then we check whether R is a generating set of $(S, \circ)$. To do so, let $S_0 = R$. We compute inductively $S_i = S_{i-1} \cup (R \circ S_{i-1})$. This adds at least one element in a step, until we reach some $k \leq n$ such that $S_k = S$. In this case, R is a set of generators. For each element a added to some set S_i, we query the $\log n$ elements $R \circ a$ to look for further elements. In total we query at most the $O(n \log n)$ elements of the $R \times S$ submatrix of T. The quantum time is bounded by $O(n^{3/2} \log n)$.

If R is a set of generators, we have to verify whether the multiplication table is associative. Light observed (see [CP61]) that if R is a set of generators of S, then it suffices to test all triples a, b, c in which b is an element of R. By using Grover search, the quantum query for finding a non-associative triple (if there is one) is $O(n\sqrt{\log n})$. By Theorem 5.5 we can decide whether this semigroup is a group. The total quantum query complexity of is $O(n \log n)$. □

The upper bound of Theorem 5.11 almost matches the lower bound we have.

Theorem 5.12 *Whether a groupoid is a quasigroup or a group requires $\Omega(n)$ quantum queries.*

Proof. We apply the Theorem 2.4. Let A be the operation table T of $\mathbb{Z}_n$ and let $\circ$ be the addition modular n. Then T is a quasigroup resp. group. The set B consists of all $n \times n$ matrices T', where one entry of T' is modified. Therefore the tables of B forming no quasigroup resp. group. The relation R is defined by

$$R = \{ (T, T') \in (A, B) \mid d(T, T') = 1 \}.$$

Then R satisfies that $m = n^2(n-1),\ m' = 1,\ l = n-1$ and $l' = 1$. Therefore the quantum query complexity to decide whether a groupoid is a quasigroup resp. group is $\Omega(n)$. □

6 The Commutativity Problem

In the commutativity problem we are given a finite set S of size n with a binary operation $\circ : S \times S \to S$ represented by a table. One has to decide whether S is a commutative. In the quantum setting, one can solve the problem in linear time by a Grover search over all tuples $(a, b) \in S^2$ that checks whether the

tuples are commutative. We show that the commutativity problem requires $\Omega(n)$ quantum queries, even when S is a monoid.

Theorem 6.1 *The quantum query complexity of the commutativity problem for groupoids, semigroups, and monoids is* $\Theta(n)$.

Proof. We start by showing the lower bound for semigroups via Theorem 2.4. Let $S = \{0, 1, \ldots, n-1\}$. The set A consists of the zero matrix of order n. The set B consists of all $n \times n$ matrices, where the entry of position (a, b) is 1, for $a \neq b \in S - \{0, 1\}$, and 0 otherwise. All operation tables of the sets A and B are semigroups. Then we have $m = \Omega(n^2)$, $m' = 1$, and the quantum query lower bound for testing if a given semigroup is commutative is $\Omega(n)$.

We reduce the commutativity problem for semigroups to the commutativity problem for monoids. Let S be a semigroup represented as a operation table T. We define a monoid $M = S \cup \{e\}$ with the identity element $e \notin S$, that is, with $a \circ e = e \circ a = a$, for all $a \in S$. Then the semigroup S is commutative iff the monoid M is commutative. □

Magniez and Nayak [MN05] quantize a classical Markov chain for testing the commutativity of a black box group given by the generators. They constructed an $O(k^{2/3} \log k)$ quantum query algorithm, where k is the number of generators of the group. In the case when $(S, \circ)$ is a quasigroup, a random set of $c \log n$ elements will be a set of generators with probability at least $1 - \exp(c)$ [RS00]. Therefore we obtain the following result:

Theorem 6.2 *Whether a quasigroup, loop or group is commutative can be decided with quantum query complexity* $O((\log n)^{\frac{2}{3}} \log \log n)$.

7 The Distributivity Problem

In the distributivity problem we are given a set S and two binary operations $\oplus : S \times S \to S$ and $\otimes : S \times S \to S$ represented by tables. One has to decide whether $(S, \oplus, \otimes)$ is distributive, i.e. we have to test whether the two equations $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ are satisfied. A triple $(a, b, c) \in S^3$ that fulfills both equations is called a *distributive triple*. In classical computing, it is not known whether this problem can be solved in less than cubic time. In the quantum setting, one can do a Grover search over all triples $(a, b, c) \in S^3$ and check whether each triple is distributive. The quantum query complexity of the search is $O(n^{3/2})$. We show a linear lower bound on the query complexity.

Theorem 7.1 *The distributivity problem requires* $\Omega(n)$ *quantum queries.*

Proof. Let $S = \{0, 1, \ldots, n-1\}$. We apply the Theorem 2.4. The set A consists of all pairs of $n \times n$ matrices $T_\oplus$ and $T_\otimes$, where $T_\otimes$ is the zero-matrix, and the entry at position $(1, 0)$ in $T_\oplus$ is 1, and 0 otherwise. It is easy to see, that the tables of A are distributive, since $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z) = 0$ for all $x, y, z \in S$. The set B consists of all pairs of $n \times n$ matrices $T'_\oplus$ and $T'_\otimes$,

where the entry of position $(1,0)$ in $T'_\oplus$, and (a,b) in $T'_\otimes$ is 1, for $a,b \in S-\{0,1\}$, and 0 otherwise. Then $a \otimes (b \oplus c) = 0$ and $(a \otimes b) \oplus (a \otimes c) = 1$ with $b \neq c$. Therefore the tables of B are not distributive.

From each $(T_\oplus, T_\otimes) \in A$, we can obtain $(T'_\oplus, T'_\otimes) \in B$ by replacing the entry 0 of $T_\otimes$ at (a,b) by 1, for any $a,b \notin \{0,1\}$. Hence we have $m = \Omega(n^2)$. From each $(T'_\oplus, T'_\otimes) \in B$, we can obtain $(T_\oplus, T_\otimes) \in A$, by replacing the entry 1 of $T_\otimes$ at position (a,b) by 0, for $a,b \notin \{0,1\}$. Thus we have $m' = 1$. By Theorem 2.4, the quantum query complexity is $\Omega(\sqrt{m \cdot m'}) = \Omega(n)$. □

If $(S, \oplus)$ is a commutative quasigroup, then we can get a faster algorithm to check distributivity. The key is that one non-distributive triple implies the existence of more such triples. Similar to Lemma 5.8, we have the following lemma.

Lemma 7.2 *Let S be a set and $\oplus, \otimes$ be two binary operations on S, such that $(S, \oplus)$ is a commutative quasigroup. If $(S, \oplus, \otimes)$ is non-distributive, then it has at least $\Omega(n)$ non-distributive triples.*

Proof. Let (a,b,c) be a non-distributive triple. Let $a = a' \oplus a''$ and consider the following cycle.

$$\begin{aligned}
(a' \oplus a'') \otimes (b \oplus c) &= ((a' \oplus a'') \otimes b) \oplus ((a' \oplus a'') \otimes c) \\
&= ((a' \otimes b) \oplus (a'' \otimes b)) \oplus ((a' \oplus a'') \otimes c) \\
&= (a' \otimes b) \oplus (a'' \otimes b) \oplus (a' \otimes c) \oplus (a'' \otimes c) \\
&= (a' \otimes b) \oplus (a' \otimes c) \oplus (a'' \otimes (b \oplus c)) \\
&= (a' \otimes (b \oplus c)) \oplus (a'' \otimes (b \oplus c)) \\
&= (a' \oplus a'') \otimes (b \oplus c).
\end{aligned}$$

Consider the case that $a \otimes (b \oplus c) \neq (a \otimes b) \oplus (a \otimes c)$ and hence, the first equation above doesn't hold. It follows that at least one of the other equations does not hold too. Therefore at least one of the following triples must be non-distributive:

1. (a', a'', b),
2. (a', a'', c),
3. (a'', b, c),
4. (a', b, c),
5. $(a', a'', b \oplus c)$.

Since $(S, \oplus)$ is a quasigroup, a can be written as $a' \oplus a''$ in n different ways. For each of these, distributivity fails in at least one of the five categories from above. Therefore there exists a category for which there are $\geq n/5$ failures.

The case that $(a \oplus b) \otimes c \neq (a \otimes c) \oplus (b \otimes c)$ can be handled similarly □

By using Lemma 7.2 in combination with the amplitude amplification (similar to Theorem 5.1) we have

Theorem 7.3 *Let $(S, \oplus)$ be a commutative quasigroup and $(S, \otimes)$ a groupoid. Whether $(S, \oplus, \otimes)$ is distributive can be decided with expected quantum query complexity of $O(n)$.*

Conclusions

In this paper we presented quantum query and time complexity bounds for several group testing problems. For a set S and a binary operation on S, we considered the decision problems whether a given structure with the promise of being a groupoid, semigroup, monoid or quasigroup is in fact a semigroup, monoid, quasigroup or a group. We also presented upper and lower bounds for testing distributivity and commutativity.

The table below summarizes the quantum query complexity (**QQC**) and the quantum time complexity (**QTC**) of the algebraic problems considered in the paper.

Problem	**Description**	**QQC**	**QTC**
Semigroup I	Decide if $\circ : S \times S \to S'$ is a semigroup for constant size of S'.	$\Omega(n)$ $O(n^{\frac{5}{4}})$	$O(n^{\frac{3}{2}} \log n)$
Semigroup II	Decide if a grupoid is a semigroup.	$\Omega(n)$ $O(n^{\frac{3}{2}})$	$O(n^{\frac{3}{2}} \log n)$
Monoid I	Decide if a groupoid is a monoid.	$\Omega(n)$ $O(n^{\frac{3}{2}})$	$O(n^{\frac{3}{2}} \log n)$
Monoid II	Decide if a semigroup is a monoid.	$O(n)$	$O(n \log n)$
Quasigroup	Decide if a groupoid is a quasigroup.	$\Omega(n)$ $O(n^{\frac{7}{6}})$	$O(n^{\frac{7}{6}} \log n)$
Group I	Decide if a groupoid is a group.	$\Omega(n)$ $O(n \log n)$	$O(n^{\frac{13}{12}} \log^c n)$
Group II	Decide if a semigroup/monoid is a group.	$O(n^{\frac{11}{14}} \log n)$	$O(n^{\frac{11}{14}} \log^c n)$
Group III	Decide if a quasigroup/loop is a group.	$\Theta(n)$	$O(n \log n)$
Commut. I	Decide if a groupoid/ semigroup/monoid is commutative.	$\Theta(n)$	$O(n \log n)$
Commut. II	Decide if a quasigroup/group is commutative.	$\widetilde{O}(\log^{\frac{2}{3}} n)$	$\widetilde{O}(\log^{\frac{2}{3}} n)$

It remains open to close the gaps between the upper and the lower bounds where they don't match. For example, some open questions are the following.

1. Is there a quantum algorithm for the semigroup problem which is better than $O(n^{1.5})$ for $|S'| = n$?

2. Is there are a classical or a quantum algorithm for the distributivity problem which is faster than the trivial bounds of $O(n^3)$ resp. $O(n^{1.5})$?

3. Are we able to prove a nontrivial lower bound for the decision problem whether a semigroup or monoid is a group?

Acknowledgments

We thank the referees of the paper from the FCT'07 and SOFSEM'08 conferences for valuable hints.

References

[Amb02] A. Ambainis, *Quantum Lower Bounds by Quantum Arguments*, Journal of Computer and System Sciences 64: pages 750-767, 2002.

[Amb03] A. Ambainis, *Quantum walks and their algorithmic applications*, International Journal of Quantum Information 1: pages 507-518, 2003.

[Amb04] A. Ambainis, *Quantum walk algorithm for element distinctness*, Proceedings of FOCS'04: pages 22-31, 2004.

[AS06] A. Ambainis, R. Špalek, *Quantum Algorithms for Matching and Network Flows*, Proceedings of STACS'06: pages 172-183, 2006.

[BBCMW01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, *Quantum lower bounds by polynomials*, Journal of ACM 48: pages 778-797, 2001.

[BBHT98] M. Boyer, G. Brassard, P. Høyer, A. Tapp, *Tight bounds on quantum searching*, Fortschritte Der Physik 46(4-5): pages 493-505, 1998.

[BCWZ99] H. Buhrman, R. Cleve, R. de Wolf, Ch. Zalka, *Bounds for Small-Error and Zero-Error Quantum Algorithms*, Proceedings of FOCS'99: pages 358-368, 1999.

[BDHHMSW01] H. Buhrman, C. Dürr, M Heiligman, P. Høyer, F. Magniez, M. Santha, R. de Wolf, *Quantum Algorithms for Element Distinctness*, Proceedings of CCC'01: pages 131-137, 2001.

[BHMT02] G. Brassard, P. Hóyer, M. Mosca, A. Tapp, *Quantum amplitude amplification and estimation*, AMS Contemporary Mathematics, Vol. 305: pages 53-74, 2002.

[BHT98] G. Brassard, P. Hóyer, A. Tapp, *Quantum Cryptanalysis of Hash and Claw-Free Functions*, Proceedings of LATIN'98: pages 163-169, 1998.

[BS06] H. Buhrman, R. Špalek, *Quantum Verification of Matrix Products*, Proceedings of SODA'06: pages 880-889, 2006.

[CEMM98] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, *Quantum algorithms revisited*, Proceedings of the Royal Society of London, Series A: pages 339-354, 1998. pages 339-354, 1998.

[CP61] A.H. Clifford, G.B. Preston, *The Algebraic Theory of Semigroups*, American Mathematical Society, 1961.

[DHHM04] C. Dürr, M. Heiligman, P. Høyer, M. Mhalla, *Quantum query complexity of some graph problems*, Proceedings of ICALP'04: pages 481-493, 2004.

[DHTW08] S. Dörn, D. Haase, J. Torán, F. Wagner, *Isomorphism and Factorization-Classical and Quantum Algorithms*, In Mathematical Analysis of Evolution, Information and Complexity, Wiley, 2008.

[Doe07a] S. Dörn, *Quantum Complexity Bounds of Independent Set Problems*, Proceedings of SOFSEM'07 (SRF): pages 25-36, 2007.

[Doe07b] S. Dörn, *Quantum Algorithms for Graph Traversals and Related Problems*, Proceedings of CIE'07: pages 123-131, 2007.

[DT07] S. Dörn, T. Thierauf, *The Quantum Query Complexity of Algebraic Properties*, Proceedings of FCT'07: pages 250-260, 2007.

[DT08a] S. Dörn, T. Thierauf, *The Quantum Complexity of Group Testing*, Proceedings of SOFSEM'08: pages 506-518, 2008.

[DT08b] S. Dörn, T. Thierauf, *On the Quantum Query Complexity of Matrix Products and the Determinant*, Preprint, 2008.

[Gro96] L. Grover, *A fast mechanical algorithm for database search*, Proceedings of STOC'96: pages 212-219, 1996.

[MN05] F. Magniez, A. Nayak, *Quantum complexity of testing group commutativity*, Proceedings of ICALP'05: pages 1312-1324, 2005.

[MNRS07] F. Magniez, A. Nayak, J. Roland, M. Santha, *Search via Quantum Walk*, Proceedings of STOC'07: pages: 575-584, 2007.

[MSS05] F. Magniez, M. Santha, M. Szegedy, *Quantum Algorithms for the Triangle Problem*, Proceedings of SODA'05: pages 1109-1117, 2005.

[NC03] M.A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2003.

[RS00] S. Rajagopalan, L. J. Schulman, *Verification of identities*, SIAM J. Computing 29(4): pages 1155-1163, 2000.

[Sim94] D.R. Simon, *On the power of quantum computation*, Proceedings of FOCS'94: pages 116-123, 1994.

[Sho94] P. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings of FOCS'94: pages 124-134, 1994.

[Sze04] M. Szegedy, *Quantum speed-up of Markov chain based algorithms*, Proceedings of FOCS'04: pages 32-41, 2004.