Quantum Query Algorithms for Automorphisms of Galois Groups

Agnis Škuškovniks
Faculty of Computing,
University of Latvia,
Raina bulvaris 19, Riga, LV-1586, Latvia
Agnis.Skuskovniks@gmail.com

Rūsiņš Freivalds
Institute of Mathematics and Computer Science,
University of Latvia,
Raina bulvaris 29, Riga, LV-1459, Latvia
Rusins.Freivalds@mii.lu.lv

***Abstract*—In this paper we study quantum query complexity of exactly (with probability 1) deciding the parity of n-permutations of numbers (from 0 to n-1). We show that for this non-Boolean problem use of quantum complexity techniques gives quite strong results as it does for other, but Boolean problems. We use Galois Theory to find new problems where quantum query algorithms are more efficient than deterministic ones.**

Keywords- quantum computin; quantum query algorithms; galois groups; permutations.

I. QUANTUM QUERY ALGORITHMS

Query algorithms are frequently used to compute Boolean functions. As an input for the query algorithm a black box oracle is used.It contains the values of the variables $x_1 = a_1, x_2 = a_2, \dots, x_n = a_n$ for known Boolean function $f(x_1, \dots, x_n)$. The values of the variable can be individually asked by the query algorithm. The results of any query can influence proceeding queries. Also the result of the algorithm is decided by taking into the account the answers received and is the value of function$f(a_1, \dots, a_n)$.

The complexity of the query algorithm is equal (by definition)to the number of the queries asked to the black box oracle. In deterministic query algorithms the next query is chosen uniquely depending only on answers previously received from the black box oracle. Probabilistic query algorithms allow randomization of the process of computation i.e. the choice of next query can depend on some randomness element.

Quantum query algorithms are formally defined in [1].For the sake of this paper we will introduce some background and parts that are necessary for our research.

Quantum query algorithm has a finite number of states. In each state it is allowed to make a query to the black box oracle and to change states in accordance to used algorithm. There are two types of algorithm steps: query operations and unitary transformations, these steps are performed in alternating order – one after the other. In the query operations step the states are divided into subsets corresponding to each query then an allowed quantum-parallel query is asked. A query "$x_i =?$" is asked in the subset states$q_{i_1}, \dots, q_{i_m}$. A pre-programmes unitary operation over the states $q_{i_1}, \dots, q_{i_m}$is known for every possible answer "$x_i = j$" . In unitary transformation stepa unitary transformation (performed by multiplying the subset state by unitary matrix) is performed transforming the amplitudes of these states accordingly. At the end of computation a measurement is performed. Measurement is a special operation in which amplitudes are transformed in probabilities. As amplitudes are complex numbers and probabilities are real, the operation is as follows: for a complex number $a + bi$ real number $a^2 + b^2$ is found. As all the operations are unitary then the total of acquired probabilities for all states equals 1. Then the states are divided in two groups: accepting states and rejecting states. The sum of accepting state probabilities is called the accepting probability. The quantum query algorithm is exact if the accepting probability is always either 1 or 0.

Deutsch and Jozsa [3], and Simon [5] introduced the notion of promise for quantum algorithm. The domain of correctness in quantum query algorithms with a promise is explicitly restricted. We are only interested in values of the algorithm that follow this restriction and do not look at other values. For instance, in this paper there is a promise on all the query algorithms that the calculated function represents a permutation.

There are many recent papers studying query algorithms that are computing Boolean functions. For reference see the survey by Buhrman and de Wolf [1].

In this paper we consider a class of functions $f(x_1, \dots, x_n)$ that are more general.

$$\{0,1,2, \dots, n-1\}^n \to \{0,1\}$$

We consider a particularly interesting case of the domain $\{0,1,2, \dots, n\}^n$– permutations.

For instance,

$$x_1 = 4, x_2 = 3, x_3 = 2, x_4 = 0, x_5 = 1$$

represents a 5 symbol $\{0,1,2,3,4\}$permutation. It can be described as permutation 43201.

For this special case of domain the functions $f: \{0,1,2, \dots, n-1\}^n \to \{0,1\}$ can be seen as representing properties of permutations. For instance, the function

$$f(0,1,2) = 1, f(1,2,0) = 1, f(2,0,1) = 1, f(0,2,1) = 0,$$
$$f(1,0,2) = 0, f(2,1,0) = 0$$

represents the property of 3 element permutation to be *even* (The other opposite property of permutation would be to be *odd*). Deciding the parity of permutation is definedasfinding whether this permutation is even or odd.

Quantum query algorithms where the black box contains a permutation were first considered by K.Iwama and

R.Freivalds [4]. Since then several authors have compared quantum and deterministic query algorithms for permutations but they were not able to find examples of properties of permutations where the complexity of quantum query algorithm is considerably smaller than the complexity of every deterministic query algorithm for the same problem.

II. Galois Groups of Polynomials

Roots of a given polynomial can be connected by different algebraic equations. For example, for two roots, A and B, the equation $A^2 + 5B^3 = 7$ can hold. Galois Theory considers roots with the property that for all the algebraic properties satisfied by the roots they still hold after permutations (or rearrangements) of the roots.

It should be noted that only algebraic equations with rational coefficients are considered. (We could specify a certain field that should contain the coefficients but, for the sake of examples below, we will use the field of rational numbers.)

All permutations for a given equation satisfying this property form a permutation group, called the Galois group of the polynomial (over the rational numbers). Consider the following examples that will illustrate this point:

The roots of quadratic equation $x^2 - 4x + 1 = 0$ can be found by using the quadratic formula. The two roots are $A = 2 + \sqrt{3}$ and $B = 2 - \sqrt{3}$ .Some of the algebraic equations that are satisfied by A and B include $A + B = 4$and $AB = 1$.

Obviously, if we exchange $A$and B, in either of these equations, we obtain another true statement. For example, the equation $A + B = 4$becomes$B + A = 4$.

To fact that this holds for every possible algebraic equation with rational coefficients relating the A and B values above (in any such equation, swapping A and B yields another true equation) is less obvious but true. The proof is done by using the theory of symmetric polynomials.

There are algebraic equations that relate A and B that are false when A and B are exchanged (for example equation $A - B - 2\sqrt{3} = 0$), however, this equation is not considered as it has the coefficient $-2\sqrt{3}$which is irrational.

We can see that the Galois group of the polynomial $x^2 - 4x + 1$ consists of two permutations: the transposition permutation which exchanges A and B and the identity permutation which leaves A and B untouched. It forms a cyclic group of order two.

Galois Theory is described in many textbooks [2].

III. Results

If a polynomial can be factored into monomialsit is said that polynomial splits over a field K. For example, $x^2 + 3x + 2$ splits in the rationals, with factors (x+2)and (x+1). Let S be a set of polynomials taken fromK[x].

The splitting field for S is thesmallest extension that splits all the polynomials in S. Call this extension F/K. If F/K is a field extension, its Galois group is the group of auto orphisms of F that fix K. If $p(x)$ is an irreducible polynomial with coefficients in K, it also has a Galois group, namely the Galois group of its splitting field. The splitting field F/K is the smallest field F that includes all the roots of $p(x)$.

Hence if $p(x)$is a polynomial with rational coefficients, then there is a smallest subfield $L(p)$ of the complex numbers $\mathbb{C}$containing both the rationals $\mathbb{Q}$ and allthe roots of $p(x)$. This is the splitting field of p over $\mathbb{Q}$. For the polynomial $p(x) = (x^2 - 2)(x^2 - 3)(x^2 - 5)$, it turns out that $L(p) = \mathbb{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5})$. Nowthis field contains, besides itself and $\mathbb{Q}$, fourteen other subfields which form a$(7;\ 3;\ 1)$ design. The varieties of the design are the seven quadratic sub fields $\mathbb{Q}(\sqrt{d})$ where $d \in \{2,3,5,6,10,15,30\}$, and the blocks are the seven biquadratic subfields $\mathbb{Q}(\sqrt{d_1}, \sqrt{d_2})$(where $d_1 d_2$ is not a perfect square).

We consider the query complexity of the problem P whether or not a given permutation of 7 elements preserves the Galois group of$(x^2 - 2)(x^2 - 3)(x^2 - 5)$.

Theorem 1.

(1) Every deterministic query algorithm for the problem P needs at least 6 queries.

(2) There exists a quantum query algorithm with 2 queries for the problem P accepting permutations in P with probability 1, and rejecting permutations not in P with probability strictly exceeding $\frac{1}{2}$.

(3) There exists a quantum query algorithm with 3 queries for the problem P accepting permutations in P with probability 1, and rejecting permutations not in P with probability 1.

Acknowledgment

The research was supported by Project 271/2012 from the Latvian Council of Science**.**

References

[1] H.Buhrman, R. de Wolf „Complexity measures and decision tree complexity: a survey.”, Theoretical Computer Science, vol. 288(1), p. 21 - 43, 2002.

[2] D.A.Cox „Galois theory.”, Wiley-Interscience, 2004.

[3] D.Deutsch, R.Jozsa „Rapid solutions of problems by quantum computation.”, Proceedings of the Royal Society of London, vol. A 439,p. 553, 1992.

[4] R.Freivalds, K.Iwama „Quantum queries on permutations with a promise.”, Lecture Notes in Computer Science, vol. 5642, p. 208 - 216, 2009.

[5] I.Simon „String matching algorithms and automata.”, Lecture Notes in Computer Science, vol. 814, p. 386 - 395, 1994.