A Package for Assembling Quantum Circuits and Generating Associated Polynomial Sets

Vladimir P. Gerdt[1], **Vasily M. Severyanov**[2]
Laboratory of Information Technologies, JINR

In [1] it is shown that elements of the unitary matrix determined by a quantum circuit can be computed by counting the number of common roots in the finite field $\mathbb{Z}_2$ for a certain set of multivariate polynomials over $\mathbb{Z}_2$. Given a quantum circuit, the polynomial set is uniquely constructed. In this paper we present a C# package called QuPol (Quantum Polynomials) that allows a user to assemble a quantum circuit and to generate the multivariate polynomial set associated with the circuit.

The generated polynomial system can further be converted into the canonical Gröbner basis form for the lexicographical monomial order. Gröbner bases form the most universal algorithmic tool of modern computer algebra to investigate and solve systems of polynomial equations [2]. Construction of the lexicographical Gröbner basis substantially alleviates the problem of the root finding for polynomial systems. To construct such a Gröbner basis, one can use efficient involutive algorithms developed in [3]. Our QuPol package together with a Gröbner basis software provides a tool for simulation of quantum circuits. We illustrate this tool by an example.

Our program has a user-friendly graphical interface and a built-in base of the elementary gates representing certain quantum gates and wires. A user can easily assemble an input circuit from those elements.

We apply the famous Feynman's sum-over-paths approach to calculate the matrix element of a quantum circuit. For this purpose we replace every quantum gate of the circuit under consideration by its classical counterpart. The trick here is to select the corresponding classical gate for the quantum Hadamard gate because for any input value, 0 or 1, it gives with equal probability either 0 or 1. We denote the output of the classical Hadamard gate by the path variable x. Its value determines one of two possible paths of computation. The classical Toffoli gate acts as $(a_1, a_2, a_3) \mapsto (a_1, a_2, a_3 \oplus a_1 a_2)$, and the classical Hadamard gate as $a_1 \mapsto x, \quad a_i, x \in \mathbb{Z}_2$.

Fig. 1 shows an example of quantum circuit (taken from [1]) and its classical correspondence. The path variables x_i comprise the (vector) path $\mathbf{x} = (x_1, x_2, x_3, x_4)^T \in \mathbb{Z}_2^4$.

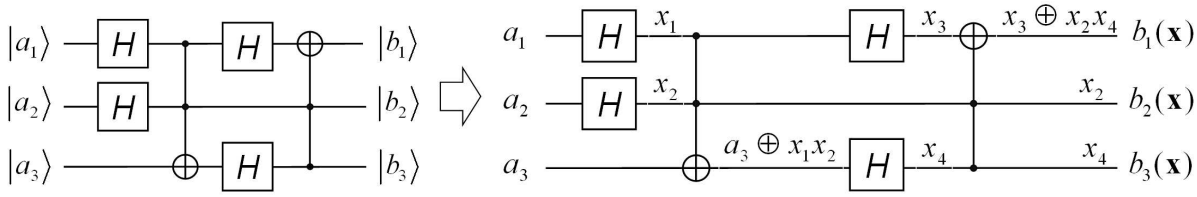

Fig. 1: From quantum to classical circuit

A classical path is a sequence of classical bit strings $a, a_1, a_2, \ldots, a_m = b$ resulting from application of the classical gates. For each selection of values for the path variables x_i

[1]E-mail: gerdt@jinr.ru
[2]E-mail: severyan@jinr.ru

we have a sequence of classical bit strings which is called an admissible classical path. Each admissible classical path has a phase which is determined by the Hadamard gates applied. The phase is changed only when the input and output of the Hadamard gate are simultaneously equal to 1, and this gives the formula

$$\varphi(\mathbf{x}) = \sum_{\text{Hadamard gates}} input \bullet output.$$

Toffoli gates do not change the phase.

For our example the phase of the path $\mathbf{x}$ is

$$\varphi(\mathbf{x}) = a_1x_1 \oplus a_2x_2 \oplus x_1x_3 \oplus x_4(a_3 \oplus x_1x_2).$$

The matrix element of a quantum circuit is given by sum over all the allowed paths from the classical states $\mathbf{a}$ to $\mathbf{b}$

$$\langle \mathbf{b}| U_f |\mathbf{a}\rangle = \frac{1}{\sqrt{2^h}} \sum_{\mathbf{x}:\mathbf{b}(\mathbf{x})=\mathbf{b}} (-1)^{\varphi(\mathbf{x})},$$

where h is the number of Hadamard gates.

Let N_0 be the number of positive terms in the sum and N_1 the number of negative terms

$$N_0 = |\{x|b(x) = b \quad \& \quad \varphi(x) = 0\}|,$$
$$N_1 = |\{x|b(x) = b \quad \& \quad \varphi(x) = 1\}|.$$

These equations count solutions to a system of $n+1$ polynomials in h variables over $\mathbb{Z}_2$. Then the matrix element may be written as the difference

$$\langle \mathbf{b}| U_f |\mathbf{a}\rangle = \frac{1}{\sqrt{2^h}} (N_0 - N_1). \tag{1}$$

For assembling arbitrary quantum circuits composed from Hadamard and Toffoli gates, we suggest to use the set of elementary gates shown on Fig. 2 and to represent a circuit as a rectangular table (Fig. 3, left) each cell of which contains an elementary gate, so that the output for each row is determined by the composition of the row elementary gates. To assemble a circuit, we define an empty table of the required size. In this case the output and phase are not fixed. Then we place required elementary gates in appropriate cells and construct the circuit polynomials (Fig. 3, right) .

A system generated by the program is a finite set $F \subset R$ of polynomials in the ring

$$R := \mathbb{Z}_2[a_i, b_j][x_1, ..., x_h], \qquad a_i, b_j \in \mathbb{Z}_2, \;\; i, j = 1, ...n,$$

in h variables and $2n$ binary coefficients. One has to count the number of roots N_0 and N_1 in $\mathbb{Z}_2$ of the polynomial sets $F_0 = \{f, ..., f_k, \varphi\}$, $F_1 = \{f, ..., f_k, \varphi + 1\}$. Then the circuit matrix is given by (1). To count the number of roots, one can convert F_0 and F_1 into a triangular form by computing the lexicographical Gröbner basis by means of the Buchberger algorithm or by involutive algorithm decribed in [3]. For the example shown on Fig. 1 we have the following polynomial system:

$$\begin{aligned} f_1 &= x_2x_4 + x_3 + b_1, \\ f_2 &= x_2 + b_2, \\ f_3 &= x_4 + b_3, \\ \varphi &= x_1x_2 + x_1x_3 + a_1x_1 + a_2x_2 + a_3x_4. \end{aligned}$$

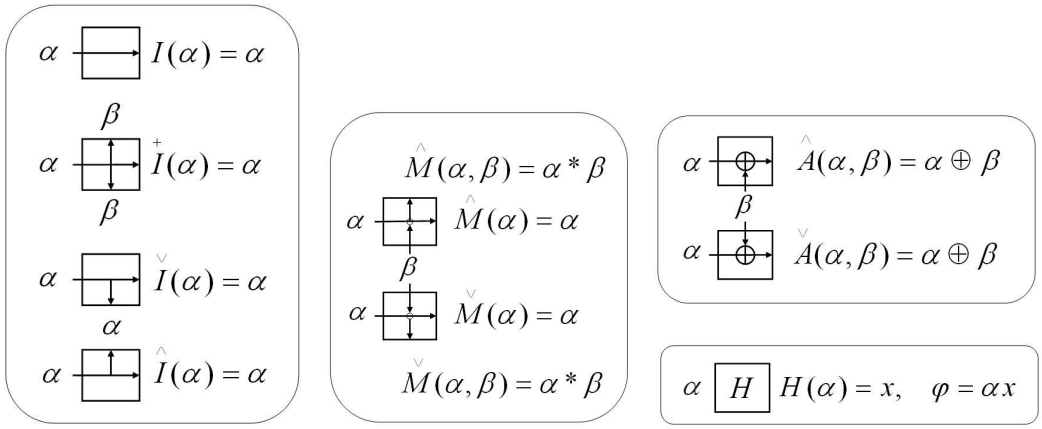

Fig. 2: Elementary gates

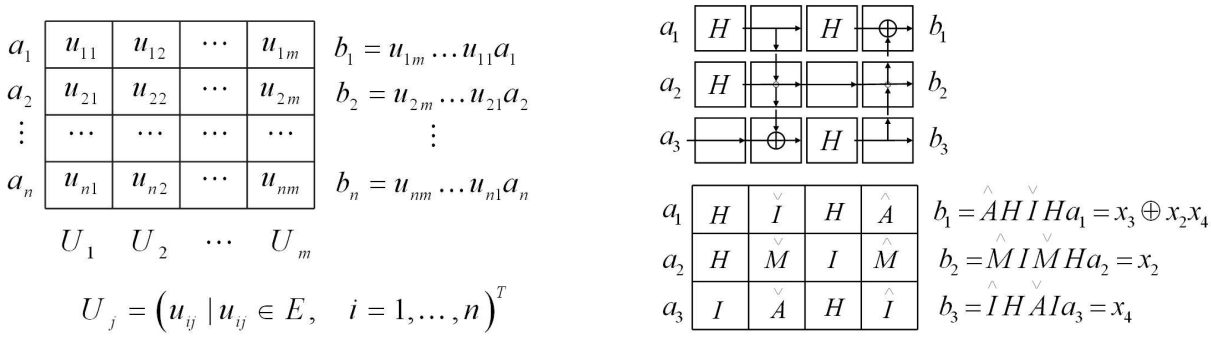

Fig. 3: Elementary decomposition (left) and assembling of a circuit (right)

The lexicographical Gröbner basis for the ordering $x_1 \succ x_2 \succ x_3 \succ x_4$ on the variables and representing both F_0 and F_1 is as follows

$$\begin{aligned} g_1 &= (a_1+b_1)x_1 + a_2b_2 + a_3b_3\ (+1),\\ g_2 &= x_2 + b_2,\\ g_3 &= x_3 + b_1 + b_2b_3,\\ g_3 &= x_4 + b_3. \end{aligned}$$

From this lexicographical Gröbner basis we immediately obtain the following conditions on the parameters:

$$\begin{aligned} a_1+b_1=0 \ \ \&\ \ a_2b_2+a_3b_3=0,\\ a_1+b_1=0 \ \ \&\ \ a_2b_2+a_3b_3=1. \end{aligned}$$

From these conditions we easily count 2 (0) roots of F_0 (F_1) and 0 (2) roots of F_0 (F_1). In all other cases there is 1 root of F_0 and F_1.

Some matrix elements are

$$\langle 000|\,U\,|000\rangle = \frac{1}{2}\ ,\quad \langle 000|\,U\,|001\rangle = -\frac{1}{2}\ ,\quad \langle 000|\,U\,|111\rangle = 0.$$

References

[1] *Christopher M. Dawson et al.* Quantum computing and polynomial equations over the finite field Z_2. *arXiv:quant-ph/0408129*, 2004.

[2] *B.Buchberger and F.Winkler (eds.) Gröbner Bases and Applications.* Cambridge University Press, 1998.

[3] *Gerdt V.P.* Involutive Algorithms for Computing Gröbner Bases. *Proceedings of the NATO Advanced Research Workshop "Computational commutative and non-commutative algebraic geometry"* (Chishinau, June 6-11, 2004), IOS Press, 2005.

[4] Aharonov D. *A Simple Proof that Toffoli and Hadamard Gatesare Quantum Universal.* arXiv:quant-ph/0301040.

[5] Microsoft Visual C# .net Standard, Version 2003.