Scientific
Research
Publishing

Formalized Operators with Phase Encoding

Nikolay Raychev

School of Computing, Varna University of Management, Varna, Bulgaria
Email: nikolay.raychev@vumk.eu

Abstract

In this article the concept of phase encoding/decoding is used to analyze and formalize a simple quantum algorithm—the Deutsch's algorithm. The algorithm is formalized in two different ways through an analysis, based on phase encoding/decoding, carried out by the formalized elementary operators developed by the author of this article. Concrete examples of different possible realizations of the formalized with Raychev's operators Deutsch's algorithms are offered.

Keywords

1. Introduction

Quantum computers can be useful in solving many important and complex issues such as solving many fundamental and complex problems such as integer factorization [1], database search [2], global binary optimization [3], linear equation solving [4], and so on [5]. The reversible computation and reversible universal logic operator was first proposed by Toffoli [6]. Based on the work of Toffoli, Bennett first implement efficient algorithm with reversiblelogic [7]. Based on the work of Toffoli and Bennett, Deutsch defines a three-qubit operator [8]. As a result of the Deutsch's work, a plurality of other universal operators [9]-[12] were found. On this basis in recent years several other complex surveys have been created [13]-[15]. This report described an approach for encoding and decoding of discrete information about basic states at the input of operators in the phase of their outputs. The decoding is viewed as a special case of interference with four different decoding forms that reflect the identity classes and negation operators. The formalized qubit operator developed by the authors [16]-[24], which is described in this research, uses the decomposition of the phase/amplitude, which characterizes the operators, presented in the report. The formalized operator permits operation with single qubit operators as linear combinations of the identity operator and the negation operator. In such case, a single qubit operator either negates, identifies, or performs partial identity/negation. If an operator decodes another one, it is able to read the information encoded in the phase space, and reduce the encoded bits to a state or its negation. Relationships of decoding

have been developed both as regards to the operator parameters and in terms of Boolean functions encoding. This in addition leads to an increase in the abstraction level. The proposed system approach is different from previous discussions for phase encoding, making the encoding a substantial part of all operators so that the correct encoded information can be determined from the operators parameters.

Both Identity$_1$ and Negation$_1$ classes are analogous to the classic operators and therefore the formalization of single qubit operators with these classes, acting as main operators, ensures means for primitive operators operation, set out in the classical concepts for calculations. The main goal is the parts of the state phase to be separated from those of the amplitude in such a way that should be set out as the key models at the interference, generated by the operators. This interference is a key to the quantum calculations and the quantum algorithms development. In addition, the separation permits the strict characterization of the consequences from a change of the phase as the binary information encoding in a phase space and thus permits the characterization in terms of Boolean functions. This report will consider the necessary and sufficient conditions for combining operators from Identity$_1$ and Negation$_1$ for formation of unitary operators and provision of an abstraction on the relative weights of the base operators. The logical formalization of the main operators will be addressed as regards to their phases and phase changes, which they carry out, on the states, to which they are applied. Whereas this approach to elementary controlled operators is elementary, the formalized system for designing of quantum circuits algorithmic models offers a second approach, which unites the elementary controlled operations with another important aspect of multitude of quantum algorithms, the Oracle operators. In the quantum algorithms design a standard technique is the use of an Oracle operator to enact a phase-kickback operation. Herein an arbitrary, probably irreversible function $f : \mathbb{B}^m \to \mathbb{B}^l$ is encoded into an $n = m + l$ qubit operator U_f, such that $U_f |xy\rangle = |x(y \oplus f(x))\rangle$, where $x \in \mathbb{B}^m$, and $y \in \mathbb{B}^l$. Usually higher degree bits for x and with a lower degree for y shall be used. The same could be performed vice versa where $U_f |yx\rangle = |(y \oplus f(x)x)\rangle$. In both cases U_f acts as an Oracle for f. Such operations are accepted as the controlled application of l qubit operator $\widehat{U_f}$, where x acts as a control bit, and y as a target bit. The phase kick-back happens when the target qubits are set to an own state $\widehat{U_f}$, and the value of $f(x)$ is encoded into the resulting state phase. It is possible this conception for the controlled operations to be applied to an elementary, two qubit controlled operators, which will provide a new means for viewing the elementary controlled operations behavior and the interference patterns that they generate. In order to generalize the behavior of a U_f-type operator, the option to occur phase shifts in addition to the main changes must be enabled. In 1985 Deutsch offers a probabilistic algorithm [5], which allows for Oracle function g: $\mathbf{In} \to \mathbf{B}$ to be calculated $g(0) \oplus g(1)$ with a probability of 1/2, using only 1 application of G: If G: $|x, y\rangle \to |x, y \oplus g(x)\rangle$ is a two qubit oracle operator, realizing the boolean oracle function g: $\mathbf{In} \to \mathbf{B}$. The register x will contain the value $g(0) \oplus g(1)$ each time, when 1 is measured in y, which in turn occurs in 50% of the cases. Although strictly speaking this does not provide an acceleration relative to the classical case, if it is taken into account that on average 2 trials are needed for the actual measurement $g(0) \oplus g(1)$, then the Deutsch's algorithm is the first evidence that the quantum computers are capable of computations in ways impossible for the classical computers. As a whole, in order to achieve any acceleration relative to the classical algorithms, it is necessary to be used the unique characteristics of the quantum computations, namely

- Superposition (step 2)
- Quantum parallelism (step 3)
- Interference (step 4)

Deutsch's algorithm has been explored many times [3] [4]. In this paper we want to focus mainly on the phase encoding and decoding with formalized Raychev's operators.

The Deutsch's algorithm points out the basic principles of the quantum algorithms, while requiring a small and relatively easy to understand circuit. Here it will be studied anew through the prism of the phase encoding and decoding and will be given more common characteristic of the algorithm. First, let's recall the algorithm and its standard presentation. The problem of Deutsch is to determine if a single bit boolean function is in *BAL* or *CONST* at given quantum external source of information, black box, to that function. It requires a single query of the source, where a classical approach would require two. **Figure 1** gives the standard circuit for Deutsch's algorithm. The Deutsch's algorithm is a key element of the introduction to the quantum computations and has been analyzed many times [3] [4]. Specifically this analysis is focused mainly on the phase encoding and decoding relationships in an attempt to generalize the construction of the circuit.

Particular attention will be paid to three elements of the Deutsch's algorithm, as it is presented in **Figure 1**.

Deutsch's algorithm

Input: True O where for the function $f: \{0, 1\} \rightarrow \{0, 1\}$ and $x, y \in \{0, 1\}$,

$O \, |x, y\rangle \rightarrow |x(y \oplus f(x))\rangle$

Preconditions: Input state to $|01\rangle$.

Output When f is a constant, then the last measured system will be $|0b\rangle$ for any $b \in \{0, 1\}$ When f is balanced, then the last measured system will be $|1b\rangle$
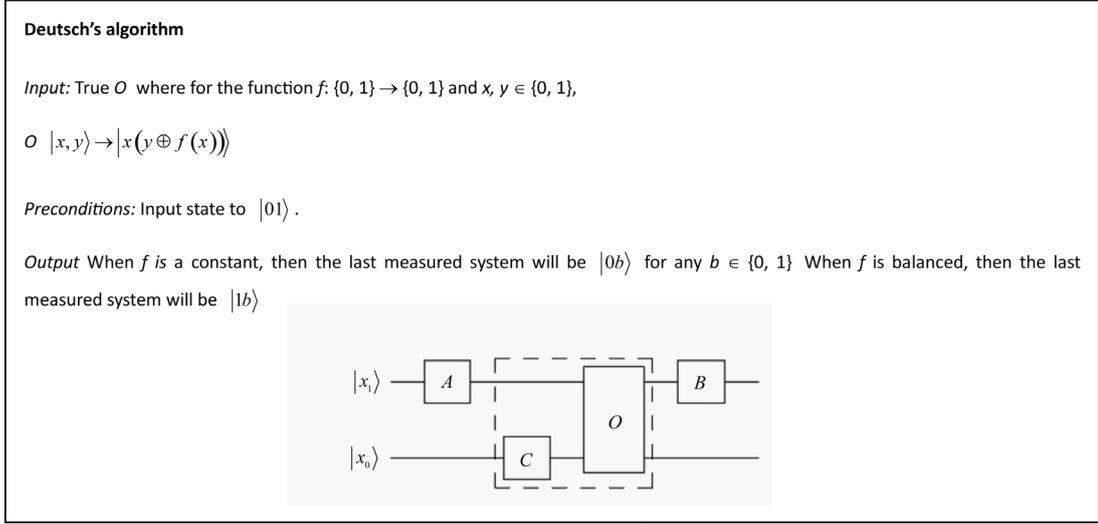
Figure 1. Algorithm of Deutsch.

1) The Hadamard operator H, which is applied to x_1 at the start and end of the algorithm, is its own identity decoder. Thus the effect of the initial and final $H_{[1]}$ can be considered as encoding and decoding of the initial state of x_1.

2) The composition of $H_{[0]}$ with an oracle operator forms an α degree $U_{B^1|_2}$ operator.

3) The selection of an initial state of the working qubit plays a role in the result of the algorithm. The conditions for the initial state of the working qubits can be expressed and explained in respect of the encoding/decoding results.

2. Formalizations of the Deutsch's Algorithms

To examine the conclusions from these observations isproposed the three-part decomposition.

The algorithm is divided into three logical steps: the application of $A_{[1]}$, followed by the Oracle operator VC, and finally the application of $B_{[1]}$ (**Figure 2**). The analysis will begin by passing through the algorithm in order to examine the state of the system with respect to the operator parameters.

If $A = U_{[1]}(\alpha_A, p_A)$, then $|\phi_1\rangle = A|x_1 x_0\rangle$, where

$$|\phi_1\rangle = A|x_1 x_0\rangle = (-1)^{\mathcal{E}(p_A)_0(x_1)} \sqrt{\alpha_A}\,|x_1 x_0\rangle + (-1)^{\mathcal{E}(p_A)_1(x_1)} \sqrt{1-\alpha_A}\,|\overline{x_1} x_0\rangle$$

If $C = U_{[0]}(\alpha_C, p_C)$ and $V = CU(1, 0, f, V_0, V_1)$, where $V_0 \in \text{Identity}_1$ and $V_1 \in \text{Negation}_1$ such that $V \in U_{B^1|_n}$.

Similar formulation of V allows the phase changes to be introduced by the Oracle operator in addition to the conditional basis changes. By Theorem 1 VC is α degree $U_{B^1|_2}$ operator. If $O = VC = CU(1, 0, f, V_0 C, V_1 C)$ and $\mathcal{E}\left(O_{f(x_1)}\right)$ denotes the phase function of $V_0 C$ when $f(x_1) = 0$, and vice versa $V_1 C$, when $f(x_1) = 1$. In this analysis the two operators of O are effectively indexed via the function f. In a similar way, the amplitude parameter of O can be indexed. If $\alpha_{f(x_1)}$ denotes α_C, when $f(x_1) = 0$, and $1 - \alpha_C$, when $f(x_1) = 1$. Then $|\phi_2\rangle = O|\phi_1\rangle$ is such that,

$$
\begin{aligned}
|\phi_2\rangle = O|\phi_1\rangle = &(-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0(x_0)} \sqrt{\alpha_A a_{f(x_1)}}\,|x_1 x_0\rangle \\
&+ (-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1(x_0)} \sqrt{\alpha_A \left(1 - a_{f(x_1)}\right)}\,|x_1 \overline{x_0}\rangle \\
&+ (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0(x_0)} \sqrt{\left(1 - \alpha_A\right) a_{f(\overline{x_1})}}\,|\overline{x_1} x_0\rangle \\
&+ (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1(x_0)} \sqrt{\left(1 - \alpha_A\right)\left(1 - a_{f(\overline{x_1})}\right)}\,|\overline{x_1} \overline{x_0}\rangle
\end{aligned}
$$

(1)
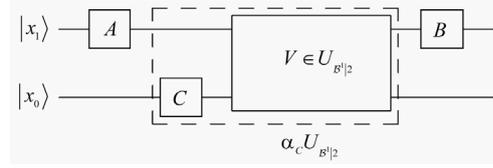
Figure 2. Formalized three-partdecomposition.

The final step of the algorithm generates interference between the operator A and B. If $B = U_{[1]}(\alpha_B, p_B)$, then $|\phi_3\rangle = B|\phi_2\rangle$. The final state of the circuit, is given in Equation (2). The formulas are rearranged, in such way that the results of the operators, generating the interference, are next to each other.

$$\langle x_1 x_0 | B | \phi_2 \rangle$$

$$= (-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}(p_B)_0(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0(x_0)} \sqrt{\alpha_A \alpha_B a_{f(x_1)}}$$

$$+ (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}(p_B)_1(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0(x_0)} \sqrt{(1-\alpha_A)(1-a_B) a_{f(\overline{x_1})}}$$

$$\langle x_1 \overline{x_0} | B | \phi_2 \rangle$$

$$= (-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}(p_B)_0(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1(x_0)} \sqrt{\alpha_A \alpha_B \left(1 - a_{f(x_1)}\right)}$$

$$+ (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}(p_B)_1(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1(x_0)} \sqrt{(1-\alpha_A)(1-a_B)\left(1 - a_{f(\overline{x_1})}\right)}$$

$$\langle \overline{x_1} x_0 | B | \phi_2 \rangle$$

$$= (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}(p_B)_0(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0(x_0)} \sqrt{(1-\alpha_A)\alpha_B a_{f(\overline{x_1})}}$$

$$+ (-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}(p_B)_1(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0(x_0)} \sqrt{\alpha_A (1-a_B) a_{f(x_1)}}$$

$$\langle \overline{x_1 x_0} | B | \phi_2 \rangle$$

$$= (-1)^{\mathcal{E}(p_A)_1(x_1) \oplus \mathcal{E}(p_B)_0(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1(x_0)} \sqrt{(1-\alpha_A) a_B \left(1 - a_{f(\overline{x_1})}\right)}$$

$$+ (-1)^{\mathcal{E}(p_A)_0(x_1) \oplus \mathcal{E}(p_B)_1(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1(x_0)} \sqrt{\alpha_A (1-a_B)\left(1 - a_{f(x_1)}\right)}$$

$$(2)$$

If the general formulas for the probability amplitude of each basis state shown in Equation (2) are given, the Deutsch's algorithm can be generalized in the context of the formalized Raychev's operators. The general character of the output data will be maintained in such way that the final state should leave x_1 unchanged, if f is constant, and change it if f is balanced. The first observation is that the final operator of the Deutsch's algorithm is a decoder of the initial operator. The decoding connections may be outlined in terms of the phase numbers, as shown in **Table 1**.

Until now the decoding was limited to the conditions under which the two operators are combined in order to form a certain basis operator. The requirements for the parameters are shown in **Table 2**.

This is done within the limits of the formalized Raychev's operators. More common structure of the decoding operators occurs when calling the main matrix formulation of the operators.

When B is a decoder of A, can be found the interference pattern, generated by B, from **Table 1**. More specifically, if B is a decoder of A, then there exists some $g, h, k \in \mathcal{B}^1$ such that $h = \overline{k}$ and $h, k \neq g$. When B is an identity decoder, g characterizes the phase change in the states that identify x_1, i.e. the states left after the decoding operation, and the functions h and k determine the interference generated in the states, where x_1 is with a negative value. Furthermore, when B is an identity decoder of A, then the amplitude parameter α_A is equivalent to α_B. From here, when B is an identity decoder of A, Equation (2) can be simplified to Equation (3).

Table 1. The negation decoding by phase number γιη.

	N	−N	X	−X
$(\gamma\eta)_A$		$\alpha_A = 1 - \alpha_B$		
000	000	011	110	101
001	010	001	100	111
010	001	010	111	100
011	011	000	101	110
100	110	101	000	011
101	100	111	010	001
110	111	100	001	010
111	101	110	011	000

Table 2. The decoding operators.

	I	Z	N	−X
$(\gamma\eta)_A$	$\alpha_A = \alpha_B$		$\alpha_A = 1 - \alpha_B$	
000	001	100	000	110
001	000	101	010	100
010	011	110	001	111
011	010	111	011	101
100	100	001	110	000
101	101	000	100	010
110	110	011	111	001
111	111	010	101	011

$$
\begin{aligned}
\langle x_1 x_0 | B | \phi_2 \rangle &= (-1)^{g(x_1) \oplus \varepsilon \left(O_{f(x_1)} \right)_0 (x_0)} \alpha_A \sqrt{a_{f(x_1)}} \\
&+ (-1)^{g(x_1) \oplus \varepsilon \left(O_{f(\overline{x_1})} \right)_0 (x_0)} (1 - \alpha_A) \sqrt{a_{f(\overline{x_1})}} \\
\langle x_1 \overline{x_0} | B | \phi_2 \rangle &= (-1)^{g(x_1) \oplus \varepsilon \left(O_{f(x_1)} \right)_1 (x_0)} \alpha_A \sqrt{\left(1 - a_{f(x_1)}\right)} \\
&+ (-1)^{g(x_1) \oplus \varepsilon \left(O_{f(\overline{x_1})} \right)_1 (x_0)} (1 - \alpha_A) \sqrt{\left(1 - a_{f(\overline{x_1})}\right)} \\
\langle \overline{x_1} x_0 | B | \phi_2 \rangle &= (-1)^{h(x_1) \oplus \varepsilon \left(O_{f(\overline{x_1})} \right)_0 (x_0)} \sqrt{\left(1 - \alpha_A\right) \alpha_A a_{f(\overline{x_1})}} \\
&+ (-1)^{k(x_1) \oplus \varepsilon \left(O_{f(x_1)} \right)_0 (x_0)} \sqrt{\alpha_A \left(1 - a_A\right) a_{f(x_1)}} \\
\langle \overline{x_1} \overline{x_0} | B | \phi_2 \rangle &= (-1)^{h(x_1) \oplus \varepsilon \left(O_{f(\overline{x_1})} \right)_1 (x_0)} \sqrt{\left(1 - \alpha_A\right) a_A \left(1 - a_{f(\overline{x_1})}\right)} \\
&+ (-1)^{k(x_1) \oplus \varepsilon \left(O_{f(x_1)} \right)_1 (x_0)} \sqrt{\alpha_A \left(1 - a_A\right) \left(1 - a_{f(x_1)}\right)}
\end{aligned}
\tag{3}
$$

If an attention is paid to the Oracle operator and how f controls its impact on the development of the system. If f is a constant function, then $f(x_1) = f(\overline{x_1})$ and Equation (3) can be further simplified, as this means that

only one of the two subspace operators, $V_0 C$ and $V_1 C$ will be used by the Oracle operator.

$$\langle x_1 x_0 | B | \phi_2 \rangle = (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} \alpha_A \sqrt{a_{f(x_1)}}$$
$$+ (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} (1-\alpha_A) \sqrt{a_{f(x_1)}}$$
$$\langle x_1 \overline{x_0} | B | \phi_2 \rangle = (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} \alpha_A \sqrt{\left(1-a_{f(x_1)}\right)}$$
$$+ (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} (1-\alpha_A) \sqrt{\left(1-a_{f(x_1)}\right)}$$
$$\langle \overline{x_1} x_0 | B | \phi_2 \rangle = (-1)^{h(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} \sqrt{\left(1-\alpha_A\right)\alpha_A a_{f(x_1)}}$$
$$+ (-1)^{k(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} \sqrt{\alpha_A \left(1-a_A\right) a_{f(x_1)}}$$
$$\langle \overline{x_1} \overline{x_0} | B | \phi_2 \rangle = (-1)^{h(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} \sqrt{\left(1-\alpha_A\right) a_A \left(1-a_{f(x_1)}\right)}$$
$$+ (-1)^{k(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} \sqrt{\alpha_A \left(1-a_A\right)\left(1-a_{f(x_1)}\right)}$$

(4)

In Equation (4) is seen that the encoding, performed by the Oracle operator, in each term of the sum of the probability amplitudes is the same. Thus, no interference is generated by the Oracle operator and the final state of the circuit is reduces down to the decoding interaction between *A and B*. The intervention carried out relative to the states $\left| \overline{x_1} x_0 \right\rangle$ and $\left| \overline{x_1 x_0} \right\rangle$, obviously leads to zero probability amplitudes when f is constant. Thus leaving the state in a superposition of $\left| x_1 x_0 \right\rangle$ and $\left| x_1 \overline{x_0} \right\rangle$. It is worth noting that achieving the interference necessary to leave x_1 unchanged at f constant, is independent from the initial state of the system. Now the impact of the Oracle operator will be considered when f is balanced. As mentioned earlier, the goal is to create a system in which the value of x_1 has changed. If f is balanced, both operators $V_0 C$ and $V_1 C$ will be used by the Oracle operator. This is implied by the fact that $f(x_1) \neq f\left(\overline{x_1}\right)$. Under the given $V_1 \in \text{Negation}_1$ and $V_1 \in \text{Identity}_1$, if α is the amplitude of $V_0 C$, then the amplitude of $V_1 C$ must be $1-\alpha$. This means that $\alpha_{f(x_1)} = 1 - \alpha_{f(\overline{x_1})}$ and allows us to further simplify Equation (3).

$$\langle x_1 x_0 | B | \phi_2 \rangle = (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} \alpha_A \sqrt{a_{f(x_1)}}$$
$$+ (-1)^{g(x_1) \oplus \varepsilon\left(O_{f(\overline{x_1})}\right)_0(x_0)} (1-\alpha_A) \sqrt{\left(1-a_{f(x_1)}\right)}$$
$$\langle x_1 \overline{x_0} | B | \phi_2 \rangle = (-1)^{g(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} \alpha_A \sqrt{\left(1-a_{f(x_1)}\right)}$$
$$+ (-1)^{g(x_1) \oplus \varepsilon\left(O_{f(\overline{x_1})}\right)_1(x_0)} (1-\alpha_A) \sqrt{a_{f(x_1)}}$$
$$\langle \overline{x_1} x_0 | B | \phi_2 \rangle = (-1)^{h(x_1) \oplus \varepsilon\left(O_{f(\overline{x_1})}\right)_0(x_0)} \sqrt{\left(1-\alpha_A\right)\alpha_A \left(1-a_{f(x_1)}\right)}$$
$$+ (-1)^{k(x_1) \oplus \varepsilon(O_{f(x_1)})_0(x_0)} \sqrt{\alpha_A \left(1-a_A\right) a_{f(x_1)}}$$
$$\langle \overline{x_1} \overline{x_0} | B | \phi_2 \rangle = (-1)^{h(x_1) \oplus \varepsilon\left(O_{f(\overline{x_1})}\right)_1(x_0)} \sqrt{\left(1-\alpha_A\right) a_A a_{f(x_1)}}$$
$$+ (-1)^{k(x_1) \oplus \varepsilon(O_{f(x_1)})_1(x_0)} \sqrt{\alpha_A \left(1-a_A\right)\left(1-a_{f(x_1)}\right)}$$

(5)

If the goal is the probability amplitudes of $\left| x_1 x_0 \right\rangle$ and $\left| x_1 \overline{x_0} \right\rangle$ to become zero when f is balanced, then it is clear that $\alpha_C = \alpha_A = \dfrac{1}{2}$ should be established such that $\alpha_{f(x_1)} = 1 - \alpha_{f(x_1)}$ and $\alpha_A = 1 - \alpha_A$. This will make the terms of the probability amplitude sums in $\left| x_1 x_0 \right\rangle$ and $\left| x_1 \overline{x_0} \right\rangle$ equivalent and will enable the interference to

reduce the final probability amplitudes to zero. From Equation (9.5) it is seen that, when

$\mathcal{E}\left(O_{f(x_1)}\right)_0(x_0) \neq \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0(x_0)$ and $\mathcal{E}\left(O_{f(x_1)}\right)_1(x_0) \neq \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1(x_0)$ then the entire necessary interference will occur; the probability amplitudes of $|x_1 x_0\rangle$ and $|x_1 \overline{x_0}\rangle$ will be inverse to each other, and all other probability amplitudes will constructively interfere. What was determined is that the desire to have $|x_1 x_0\rangle$ and $|x_1 \overline{x_0}\rangle$ destructively interfere sets many of the requirements for correctness of the algorithm. The desired interference can be guaranteed regardless of the value of x_0, if an Oracle operator O is defined such that

$O = CU\left(1, 0, f, U\left(\frac{1}{2}, \gamma\iota\eta\right), U\left(\frac{1}{2}, \gamma\iota\overline{\eta}\right)\right)$. This construction is not intuitive, as it requires the Oracle operator V

to be defined not as the $U_{f|2}$ Oracle typical for the Deutsch's algorithm, but as an operator in $U_{ZERO|2}$, which carries out a global phase shift that is conditionally upon the function f. The conditional phase shifts are used in the Grover's algorithm and therefore this particular operator is not something new in the quantum computations [5].

2.1. Formalized Deutsch's Algorithm: Version 1

Problem: Upon given Oracle like operator $O = CU\left(1, 0, f, I, -I\right)$, to be determined whether $f \in \mathcal{B}^1$ is constant or balanced.

Preconditions:

1) To be defined B and A such that $BA \in Ext_1$ and the amplitude parameter of A is $\alpha_A = \frac{1}{2}$.

2) $C = U\left(\frac{1}{2}, p_C\right)$

Result: If the function f is balanced $\sum_{b \in \mathbb{B}} P\left[|x_1 x_0\rangle \to |\overline{x_1} b\rangle\right] = 1$, and otherwise, *i.e.* when f is constant, then $\sum_{b \in \mathbb{B}} P\left[|x_1 x_0\rangle \to |x_1 b\rangle\right] = 1$ (**Figure 3**).

For Boolean function $f \in \mathcal{B}^m$, where $m < n$, the generalized controlled operator can be used to represent operators $U_{\mathcal{B}^m|n}$, as well as α degree version of these operators. Theorem 1 describes the construction of an operator in the set $U_{\mathcal{B}^m|n}$ whereas Theorem 2 shows the construction of an α degree version of a $U_{\mathcal{B}^m|n}$ operator.

Theorem 1 *If operator $V = CU\left(C, t, f, A, B\right)$ is such that $A \in$ The set of the identity formalized Raychev's operators, $B \in$ The set of the negation formalized Raychev's operators and $f \in \mathcal{B}^m$, where $m < n$. Then $V \in U_{\mathcal{B}^m|n}$.*

Proof. If it is assumed that $A \in$ The set of the identity formalized Raychev's operators and $B \in$ The set of the negation formalized Raychev's operators, therefore for each n qubit basis $|x\rangle$,

$$V|x\rangle = \begin{cases} \pm|x\rangle & f(C) = 0 \\ \pm|x \oplus 2^t\rangle & f(C) = 1 \end{cases} = \pm|x \oplus f(C) 2^t\rangle \tag{6}$$

Theorem 2 *If operator $V = CU\left(C, t, f, A, B\right)$ is such that $A = \left(\alpha, p_A\right)$ and $B = U\left(1 - \alpha, p_B\right)$. Then V is an α degree $U_{\mathcal{B}^m|n}$.*

Proof. If it is assumed that $A = \left(\alpha, p_A\right)$ and $B = U\left(1 - \alpha, p_B\right)$. Then, for each n qubit basis $|x\rangle$,
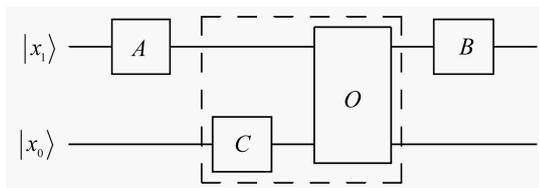
Figure 3. Formalized Deutsch's algorithm: Version 1.

$$V|x\rangle = \begin{cases} (-1)^{\mathcal{E}(p_A)_0(x_t)} \sqrt{\alpha} \,|\, x + (-1)^{\mathcal{E}(p_A)_1(x_t)} \sqrt{1-\alpha} \,|\, x \oplus 2^t \rangle & f(C) = 0 \\ (-1)^{\mathcal{E}(p_B)_0(x_t)} \sqrt{\alpha} \,|\, x \oplus 2^t \rangle + (-1)^{\mathcal{E}(p_B)_1(x_t)} \sqrt{1-\alpha} \,|\, x \rangle & f(C) = 1 \end{cases} \qquad (7)$$

$$= \pm\sqrt{\alpha} \,|\, x \oplus f(C) 2^t \rangle + \pm\sqrt{1-\alpha} \,|\, x \oplus \overline{f}(C) 2^t \rangle$$

Theorem 1 leads to general means for constructing of α degree $U_{\mathcal{B}^m|n}$ operators from an elementary indexed operator and an operator in $U_{\mathcal{B}^m|n}$.

Theorem 3 *If V is an n qubit $U_{\mathcal{B}^m|n}$ operator with target bit t and $A_{[t]}$ is indexed, formalized operator with an amplitude parameter α, such that $0 < \alpha < 1$. Then the operator VA is α degree $U_{\mathcal{B}^m|n}$ operator.*

Proof. The proof follows from theorems 1 and 2, by noting that when $V = CU(C, t, f, U, W)$, then $VA = CU(C, t, f, UA, WA)$.

Theorem 3 is important because, it captures a common occurrence in the quantum algorithms: setting the target bit of an Oracle operator to a superposition and then applying the oracle operator. In the formalized system for designing of algorithmic models for quantum circuits this can be addressed in the context of an *α degree* $U_{\mathcal{B}^m|n}$ operator.

The composition of random formalized operators with main operators can be expressed in terms of the transformations of parameters.

Formal prerequisite 1 *If the operator $A = U(\alpha, \gamma\iota\eta)$. Then*

$$\mathcal{T}_{00} A = A\mathcal{T}_{00} = A$$

$$\mathcal{T}_{01} A = A\mathcal{T}_{01} = -A$$

$$\mathcal{N}_{00} A = U((1-\alpha), \gamma\iota\overline{\eta})$$

$$\mathcal{T}_{10} A = U(\alpha, \overline{\gamma}\iota\overline{\eta})$$

$$\mathcal{N}_{10} A = U((1-\alpha), \overline{\gamma}\eta\iota)$$

$$A\mathcal{T}_{10} = U(a, \overline{\gamma}\eta\iota)$$

$$A\mathcal{N}_{10} = U((1-\alpha), \gamma(\overline{\gamma} \oplus \eta)(\gamma \oplus \iota)) \qquad (8)$$

Proof. If *A* is a base operator. Then

$$A \circ U(a, \gamma\iota\eta) = A \circ \left(\sqrt{a}\mathcal{T}_{\gamma\eta} + \sqrt{1-a}\mathcal{N}_{\gamma\eta}\right) = \sqrt{a}A \circ \mathcal{T}_{\gamma\eta} + \sqrt{1-a}A \circ \mathcal{N}_{\gamma\eta}$$

If the consideration is limited only to the original formulation of the problem, *i.e.* $U_{f|2}$ Oracle only, then the problem of Deutsch still can be solved deterministically, by choosing an appropriate starting value for x_0, the target big of U_f and *C*. The amplitude parameter of *C* can be set to $\frac{1}{2}$. By Theorem 3, the oracle operator $V \in U_{f|n}$ is combined with *C*, *to form an* $\alpha_C U_{f|n}$ Oracle. At $V = CU(1, 0, f, V_0, V_1) \in U_{f|n}$ the subspace operators of the composite operator $VC_{[0]}$ are $V_0 C = U\left(\frac{1}{2}, p_{C'}\right)$ and $V_1 C = U\left(\frac{1}{2}, p_{C''}\right)$. The exact phase of $V_1 C$ and $V_0 C$ is determined through Theorem 4.

Remark *The generalization of the Deutsch's algorithm given in Figure 2, also covers the case when are used classical, i.e. without a phase change, oracle operators of the form $CU(1, 0, f, I, X)$ as the ones used in the standard version of the Deutsch's algorithm. Then the precondition for the value of x_0 is limited to a condition on the phase of operator C and the initial value of x_0 only.*

2.2. Formalized Deutsch's Algorithm: Version 2

Problem: Upon given Oracle operator $O = CU(1, 0, f, V_0, V_1) \in U_{f|n}$, should be determined whether $f \in \mathcal{B}^1$ is constant or balanced.

Preconditions:

1) To be defined B and A such that $BA \in Ext_1$ and the amplitude parameter of A is $\alpha_A = \dfrac{1}{2}$.

2) $C = U\left(\dfrac{1}{2}, p_C\right)$

3) For operators $V_0 C = U\left(\dfrac{1}{2}, p_{C'}\right)$ and $V_1 C = U\left(\dfrac{1}{2}, p_{C''}\right)$, should be chosen initial value for x_0 such that $\mathcal{E}(p_{C'})_0(x_0) \neq \mathcal{E}(p_{C''})_0(x_0)$ and $\mathcal{E}(p_{C'})_1(x_0) \neq \mathcal{E}(p_{C''})_1(x_0)$.

Result: If the function f is balanced $\sum_{b \in \mathbb{B}} P\left[\left|x_1 x_0\right\rangle \to \left|\overline{x_1}b\right\rangle\right] = 1$, and otherwise, *i.e.* when f is constant, then $\sum_{b \in \mathbb{B}} P\left[\left|x_1 x_0\right\rangle \to \left|x_1 b\right\rangle\right] = 1$ (**Figure 4**).

If $p_C = \gamma_1 \eta$. Then should be chosen x_0 such that $\mathcal{E}(\gamma_1 \eta)_0(x_0) \neq \mathcal{E}(\overline{\gamma_1}\eta)_0(x_0)$ and $\mathcal{E}(\gamma_1 \eta)_1(x_0) \neq \mathcal{E}(\overline{\gamma_1}\eta)_1(x_0)$.

Two generalized algorithms for solving the Deutsch's problem are developed, by re-examining anew the standard formulation of the Deutsch's algorithm. In both cases are used the specific phase encoding and decoding ideas, realized by the developed by the author of the report formalized operators. Both algorithms set the initial state to be equal to a superposition of all possible two qubit states and in this way set the working bit x_0 to an own state of the Oracle operator. Furthermore, the use of Hadamard-like gates with amplitude parameters $\dfrac{1}{2}$ is, in general, a requirement of the algorithm if deterministic results are sought.

Finally, the formalizations of the Deutsch's algorithm, given in **Figure 3** and **Figure 4**, are reduced to the ability to work in many different phases, possible within the quantum computations. Perhaps the most interesting result of the analysis, based on the phase decoding, is the reaching to formalization of the algorithm in **Figure 1** in the style of Grover. This version will lead to results, which would be expected upon solving the Deutsch's algorithm, independently of its input data, but requires an Oracle operator for phase shifting as opposed to the traditional formulation of the Oracle operator in the Deutsch's problem. The result is inherited by the explicit characterization of the behavior upon phase encoding / decoding of the operators, based on their formalization.

2.3. Examples of Formalized Algorithm for Solving the Deutsch's Problem

The first example deals with a slight variation of the traditional Deutsch's algorithm.

Let operator $A = U\left(\dfrac{1}{2}, 000\right)$ with identity decoder $B = U\left(\dfrac{1}{2}, 100\right)$ such that $BA = Z$. If defines Oracle operator $O = VC$, where $C = U_{[0]}\left(\dfrac{1}{2}, 100\right)$ and $V = CU(1, 0, f, I, X)$. When the target qubit $x_0 = 1$, O and preparation of target qubit are equivalent to traditional Deutsch's algorithm. From Equation (2) can determine the final state of the system, if the input data are given for the traditional algorithm $\left|01\right\rangle$.

$$\left\langle 01\right| B_{[1]} O A_{[1]} \left|01\right\rangle = (-1)^{ZERO(0) \oplus ID(0) \oplus \mathcal{E}\left(O_{f(0)}\right)_0(1)} \frac{1}{2}^{3/2} + (-1)^{ID(0) \oplus ZERO(1) \oplus \mathcal{E}\left(O_{f(1)}\right)_0(1)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(0) \oplus \mathcal{E}\left(O_{f(0)}\right)_0(1)} \frac{1}{2}^{3/2} + (-1)^{ID(0) \oplus \mathcal{E}\left(O_{f(1)}\right)_0(1)} \frac{1}{2}^{3/2}$$
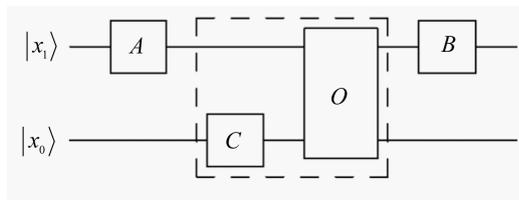
Figure 4. Formalized Deutsch's algorithm: Version 2.

$$\langle 00|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ZERO(0)\oplus ID(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_1(1)}\frac{1}{2}^{3/2} + (-1)^{ID(0)\oplus ZERO(1)\oplus \mathcal{E}\left(O_{f(1)}\right)_1(1)}\frac{1}{2}^{3/2}$$

$$= (-1)^{ID(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_1(1)}\frac{1}{2}^{3/2} + (-1)^{ID(0)\oplus \mathcal{E}\left(O_{f(1)}\right)_1(1)}\frac{1}{2}^{3/2}$$

$$\langle 11|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ID(0)\oplus ID(1)\oplus \mathcal{E}\left(O_{f(1)}\right)_0(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus ZERO(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_0(1)}\frac{1}{2}^{3/2}$$

$$= (-1)^{ONE(0)\oplus \mathcal{E}\left(O_{f(1)}\right)_0(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_0(1)}\frac{1}{2}^{3/2}$$

$$\langle 10|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ID(0)\oplus ID(1)\oplus \mathcal{E}\left(O_{f(1)}\right)_1(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus ZERO(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_1(1)}\frac{1}{2}^{3/2}$$

$$= (-1)^{ONE(0)\oplus \mathcal{E}\left(O_{f(1)}\right)_0(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus \mathcal{E}\left(O_{f(0)}\right)_0(1)}\frac{1}{2}^{3/2}$$

When f is constant, then the phase encoding executed by Oracle operator is the same in both terms of any amount. Without loss of generality, let us assume that $f = ZERO$, then $\mathcal{E}\left(O_{f(1)}\right) = \mathcal{E}\left(O_{f(0)}\right) = (ID, ZERO)$. Thus is obtained the following result

$$\langle 01|B_{[1]}OA_{[1]}|01\rangle = -\frac{1}{2}^{3/2} - \frac{1}{2}^{3/2} = -\sqrt{\frac{1}{2}}$$

$$\langle 00|B_{[1]}OA_{[1]}|01\rangle = \frac{1}{2}^{3/2} + \frac{1}{2}^{3/2} = \sqrt{\frac{1}{2}}$$

$$\langle 11|B_{[1]}OA_{[1]}|01\rangle = \frac{1}{2}^{3/2} - \frac{1}{2}^{3/2} = 0$$

$$\langle 10|B_{[1]}OA_{[1]}|01\rangle = -\frac{1}{2}^{3/2} + \frac{1}{2}^{3/2} = 0$$

When f is balanced then $C_{[0]}$, with encoding function $(ID, ZERO)$, is applied when $f(x_1) = 0$, and $(XC)_{[0]} = U_{[0]}\left(\frac{1}{2}, 000\right)$, with encoding function $(ZERO, ID)$ is applied when $f(x_1) = 1$. It should be noted that for $x_0 = 1 ID(x_0) \neq ZERO(x_0)$ and preconditions are met. Thus, when $f = ID$,

$$\langle 01|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ZERO(0)\oplus ID(0)\oplus ID(1)}\frac{1}{2}^{3/2} + (-1)^{ID(0)\oplus ZERO(1)\oplus ZERO(1)}\frac{1}{2}^{3/2} = -\frac{1}{2}^{3/2} + \frac{1}{2}^{3/2} = 0$$

$$\langle 00|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ZERO(0)\oplus ID(0)\oplus ZERO(1)}\frac{1}{2}^{3/2} + (-1)^{ID(0)\oplus ZERO(1)\oplus ID(1)}\frac{1}{2}^{3/2} = \frac{1}{2}^{3/2} - \frac{1}{2}^{3/2} = 0$$

$$\langle 11|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ID(0)\oplus ID(1)\oplus ZERO(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus ZERO(0)\oplus ID(1)}\frac{1}{2}^{3/2} = -\frac{1}{2}^{3/2} - \frac{1}{2}^{3/2} = -\sqrt{\frac{1}{2}}$$

$$\langle 10|B_{[1]}OA_{[1]}|01\rangle = (-1)^{ID(0)\oplus ID(1)\oplus ID(1)}\frac{1}{2}^{3/2} + (-1)^{ZERO(0)\oplus ZERO(0)\oplus ZERO(1)}\frac{1}{2}^{3/2} = -\frac{1}{2}^{3/2} - \frac{1}{2}^{3/2} = -\sqrt{\frac{1}{2}}$$

If we consider the example of the algorithm represented in **Figure 1**. Let operator $A = U\left(\frac{1}{2}, 000\right)$ with identity decoder $B = U\left(\frac{1}{2}, 001\right)$ such that $BA = I$. If it defines such Oracle operator $O = CU(1, 0, f, I, -I)$, such that for each operator $C = U\left(\frac{1}{2}, 010\right)$, $OC = CU\left(1, 0, f, U\left(\frac{1}{2}, 010\right), U\left(\frac{1}{2}, 001\right)\right)$. It is seen that for any

input $\left|x_1 x_0\right\rangle$ this algorithm will have the same result as Deutsch's Algorithm.

$$\left\langle x_1 x_0 \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ZERO(x_1) \oplus ZERO(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1) \oplus NOT(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0 (x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ZERO(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0 (x_0)} \frac{1}{2}^{3/2}$$

$$\left\langle x_1 \overline{x_0} \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ZERO(x_1) \oplus ZERO(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1) \oplus NOT(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1 (x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ZERO(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1 (x_0)} \frac{1}{2}^{3/2}$$

$$\left\langle \overline{x_1} x_0 \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ID(x_1) \oplus ZERO(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus NOT(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0 (x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_0 (x_0)} \frac{1}{2}^{3/2} + (-1)^{NOT(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_0 (x_0)} \frac{1}{2}^{3/2}$$

$$\left\langle \overline{x_1 x_0} \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ID(x_1) \oplus ZERO(\overline{x_1}) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1 (x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus NOT(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1 (x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(x_1) \oplus \mathcal{E}\left(O_{f(\overline{x_1})}\right)_1 (x_0)} \frac{1}{2}^{3/2} + (-1)^{NOT(x_1) \oplus \mathcal{E}\left(O_{f(x_1)}\right)_1 (x_0)} \frac{1}{2}^{3/2}$$

Without loss of generality, let us assume that for a balanced f, $\mathcal{E}\left(O_{f(x_1)}\right) = (ONE, ID)$ and $\mathcal{E}\left(O_{f(\overline{x_1})}\right) = (ZERO, NOT)$ such that they correspond to the phases of C and $-C$, respectively Oracle subspace operators. That leaves the following final state.

$$\left\langle x_1 x_0 \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ZERO(x_1) \oplus ZERO(x_1) \oplus ONE(x_0)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1) \oplus NOT(\overline{x_1}) \oplus ZERO(x_0)} \frac{1}{2}^{3/2}$$

$$= -\frac{1}{2}^{3/2} + \frac{1}{2}^{3/2}$$

$$\left\langle x_1 \overline{x_0} \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ZERO(x_1) \oplus ZERO(x_1) \oplus ID(x_0)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1) \oplus NOT(\overline{x_1}) \oplus NOT(x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(x_0)} \frac{1}{2}^{3/2} + (-1)^{NOT(x_1)} \frac{1}{2}^{3/2}$$

$$\left\langle \overline{x_1} x_0 \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ID(x_1) \oplus ZERO(\overline{x_1}) \oplus ZERO(x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus NOT(x_1) \oplus ONE(x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(x_1)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1)} \frac{1}{2}^{3/2}$$

$$\left\langle \overline{x_1 x_0} \left| B_{[1]} OA_{[1]} \right| x_1 x_0 \right\rangle = (-1)^{ID(x_1) \oplus ZERO(\overline{x_1}) \oplus NOT(x_0)} \frac{1}{2}^{3/2} + (-1)^{ZERO(x_1) \oplus NOT(x_1) \oplus ID(x_0)} \frac{1}{2}^{3/2}$$

$$= (-1)^{ID(x_1) \oplus NOT(x_0)} \frac{1}{2}^{3/2} + (-1)^{ID(x_1) \oplus NOT(x_0)} \frac{1}{2}^{3/2}$$

In the case where f is a constant, there is the same overall result as the standard formulation of the Deutch's algorithm; the Oracle operator makes no interference between the subspace and the calculation is reduced to the decoding effect B has on A.

3. Conclusions

Figure 1 and Figure 2 present two formalizations of the Deutsch's algorithms. The first requires Oracle opera-

tors, oriented to phase change, while the second formalization effectively establishes, how different operator phases can be used in the context of the standard structure of the Deutsch's algorithm. The analysis which leads to the Formalizations, strongly leans to the close relationships and properties of the boolean encoding functions of the formalized Raychev's operators, included in the algorithm.

The first version of formalized algorithm uses phase change operations where the second version demonstrates how different phases might be used in the context of the conventional Deutsch's algorithm.

The formalized Raychev's operators separate the parts of phase and amplitude and allow for expression of amplitude in terms of probabilities as opposed to more general probability amplitudes. This formalization is further improved by the characterization of classes with defined relations and incarnation of formalized logic parameter γ on the global phase. Under the needed and sufficient conditions to construct a unitary, in this case orthogonal operators are incorporated into the formalization.

References

[1] Shor, P. (1997) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, **26**, 1484-1509. http://dx.doi.org/10.1137/S0097539795293172

[2] Childs, A.M., Landahl, A.J. and Parrilo, P.A. (2007) Quantum Algorithm for Ordered Search Problem via Semidenite Programming. *Physical Review A*, **75**, Article ID: 032335. http://dx.doi.org/10.1103/PhysRevA.75.032335

[3] Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A. and Preda, D. (2001) A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, **292**, 472-475. http://dx.doi.org/10.1126/science.1057726

[4] Harrow, A.W., Hassidim, A. and Lloyd, S. (2009) Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.*, **103**, 150502. http://dx.doi.org/10.1103/PhysRevLett.103.150502

[5] Bacon, D. and van Dam, W. (2010) Recent Progress in Quantum Algorithms. *Commun. ACM*, **53**, 84-93.

[6] Toffoli, T. (1980) Reversible Computing. In: *Proceedings of the* 7*th Colloquium on Automata*, *Languages and Programming*, Springer-Verlag, London, 632-644. http://dx.doi.org/10.1007/3-540-10003-2_104

[7] Bennett, C.H. (1989) Time/Space Trade-Offs for Reversible Computation. *SIAM Journal on Computing*, **18**, 766-776. http://dx.doi.org/10.1137/0218053

[8] Deutsch, D. (1989) Quantum Computational Networks. *Proceedings of the Royal Society of London A*, **425**, 73-90. http://dx.doi.org/10.1098/rspa.1989.0099

[9] Barenco, A. (1995) A Universal Two-Bit Gate for Quantum Computation. *Proceedings ofthe Royal Society of London A*, **449**, 679-683.

[10] Deutsch, D., Barenco, A. and Ekert, A. (1995) Universality in Quantum Computation. *Proceedings of the Royal Society of London*. *Series A*: *Mathematical and Physical Sciences*, **449**, 669-677. http://dx.doi.org/10.1098/rspa.1995.0065

[11] Di Vincenzo, D.P. (1995) Two-Bit Gates Are Universal for Quantum Computation. *Physical Review A*, **51**, 1015-1022. http://dx.doi.org/10.1103/PhysRevA.51.1015

[12] Lloyd, S. (1995) Almost Any Quantum Logic Gate Is Universal. *Physical Review Letters*, **75**, 346-349. http://dx.doi.org/10.1103/physrevlett.75.346

[13] Nielsen, M. and Chuang, I. (2003) Quantum Computing and Quantum Information. Cambridge University Press, Cambridge.

[14] Phillip Kaye, R.L. and Mosca, M. (2007) An Introduction to Quantum Computing. Oxford University Press, Oxford.

[15] Rieffel, E. and Polak, W. (1998) An Introduction to Quantum Computing for Non-Physicists. http://arxiv.org/abs/quant-ph/9809016

[16] Raychev, N. (2015) Unitary Combinations of Formalized Classes in Qubit Space. *International Journal of Scientific and Engineering Research*, **6**, 395-398. http://dx.doi.org/10.14299/ijser.2015.04.003

[17] Raychev, N. (2015) Functional Composition of Quantum Functions. *International Journal of Scientific and Engineering Research*, **6**, 413-415. http://dx.doi.org/10.14299/ijser.2015.04.004

[18] Raychev, N. (2015) Logical Sets of Quantum Operators. *International Journal of Scientific and Engineering Research*, **6**, 391-394. http://dx.doi.org/10.14299/ijser.2015.04.002

[19] Raychev, N. (2015) Controlled Formalized Operators. *International Journal of Scientific and Engineering Research*, **6**, 1467-1469. http://dx.doi.org/10.14299/ijser.2015.05.003

[20] Raychev, N. (2015) Controlled Formalized Operators with Multiple Control Bits. *International Journal of Scientific and Engineering Research*, **6**, 1470-1473. http://dx.doi.org/10.14299/ijser.2015.05.001

[21] Raychev, N. (2015) Connecting Sets of Formalized Operators. *International Journal of Scientific and Engineering Research*, **6**, 1474-1476. http://dx.doi.org/10.14299/ijser.2015.05.002

[22] Raychev, N. (2015) Indexed Formalized Operators for N-Bit Circuits. *International Journal of Scientific and Engineering Research*, **6**, 1477-1480.

[23] Raychev, N. (2015) Universal Quantum Operators. *International Journal of Scientific and Engineering Research*, **6**, 1369-1371. http://dx.doi.org/10.14299/ijser.2015.06.005

[24] Raychev, N. (2015) Encoding and Decoding of Additional Logic in the Phase Space of All Operators. *International Journal of Scientific and Engineering Research*, **6**, 1356-1366. http://dx.doi.org/10.14299/ijser.2015.07.003