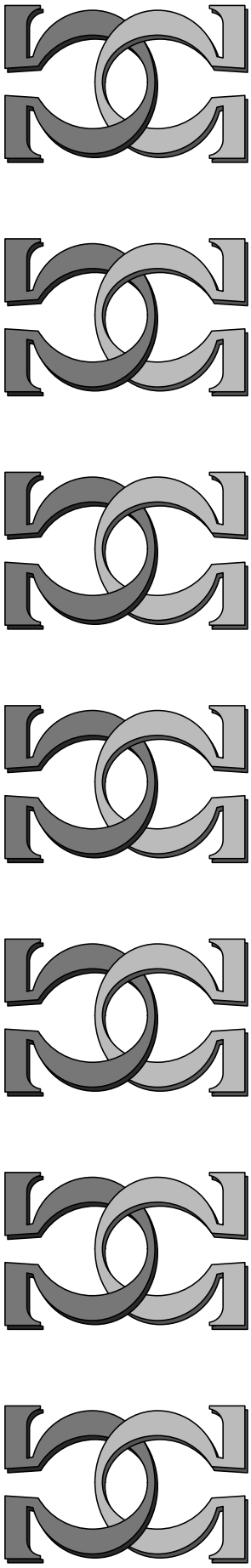
An Observer-Based De-Quantisation of Deutsch's Algorithm



CDMTCS
Research
Report
Series

An Observer-Based De-Quantisation of Deutsch's Algorithm

C. S. Calude[1], **M. Cavaliere**[2], **R. Mardare**[3]
[1]University of Auckland, NZ
[2]CNB - CSIC, Madrid, Spain
[3]Microsoft Research-University of Trento, Italy

CDMTCS-382
May 2010; revised September 2010

Centre for Discrete Mathematics and
Theoretical Computer Science

International Journal of Foundations of Computer Science

An Observer-Based De-Quantisation of Deutsch's Algorithm

Cristian S. Calude*
Department of Computer Science
University of Auckland, New Zealand
cristian@cs.auckland.ac.nz
www. cs. auckland. ac. nz/ ~ cristian

Matteo Cavaliere
CNB - CSIC
Madrid, Spain
mcavaliere@cnb.csis.es

Radu Mardare
Microsoft Research-University of Trento, Italy
mardare@cosbi.eu

Received September 28, 2010
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

Deutsch's problem is the simplest and most frequently examined example of computational problem used to demonstrate the superiority of quantum computing over classical computing. Deutsch's quantum algorithm has been claimed to be faster than any classical ones solving the same problem, only to be discovered later that this was not the case. Various de-quantised solutions for Deutsch's quantum algorithm—classical solutions which are as efficient as the quantum one—have been proposed in the literature. These solutions are based on the possibility of classically simulating "superpositions", a key ingredient of quantum algorithms, in particular, Deutsch's algorithm. The de-quantisation proposed in this note is based on a classical simulation of the quantum measurement achieved with a model of observed system. As in some previous de-quantisations of Deutsch's quantum algorithm, the resulting de-quantised algorithm is deterministic. Finally, we classify observers (as finite state automata) that can solve the problem in terms of their "observational power".

Keywords: Deutsch problem, Deutsch's quantum algorithm, de-quantisation

2010 Mathematics Subject Classification: 68W01,81P68

*Corresponding author.

1. Introduction

The "brute-force" classical simulation of a quantum algorithm—derived from the matrix mechanics formulation of quantum mechanics [13]—increases exponentially the computational time. Is it possible to do it better? The answer is affirmative. The de-quantisation of a quantum algorithm is a technique to develop a classical algorithm which: a) solves the same problem as the given quantum algorithm, b) is not exponentially slower in time compared to the quantum algorithm. The paper [2] reviews the main techniques and results in de-quantisation.

Why de-quantisation? Quantum algorithms are notoriously difficult to run, so a quantum algorithm would be preferred to a classical algorithm only if the quantum algorithm is provable faster than any classical one solving the same problem. For most known quantum algorithms such results are not available. Understanding the conditions when de-quantisation is impossible reveals features that are necessary for a quantum algorithm to be faster than any classical one. Conversely, successful de-quantisations produce efficient classical algorithms designed on radically new techniques inspired from quantum computation.

Deutsch's problem is the simplest and most frequently examined example of computational problem used to show the power and superiority of quantum computing over classical computing [11, 10, 14, 16, 4]. De-quantised solutions for Deutsch's quantum algorithm have been proposed in the literature [3, 4, 15, 1]. These solutions are based on the possibility of efficiently simulating "superpositions". In this note we take a different approach: we focus on the interplay between an observed system and its observer. More precisely, we use a model of observed system[a] to present an observer-based de-quantisation of the Deutsch's quantum algorithm which allows us to investigate the role of the "power" of the external observer. As in some previous studies [4, 1, 2], our de-quantised algorithm is deterministic and produces more information than Deutsch's quantum algorithm.

2. Automata theory preliminaries

We use some basic notions from automata theory and formal languages [17]. By V^* we denote the set of strings over the alphabet V; λ is the empty string and $V^+ = V^* \setminus \{\lambda\}$. The concatenation of the strings w_1 andw_2 is denoted by w_1w_2.

A *finite state automaton* (FSA), with no final states, is a 4-tuple $\mathbb{A} = (\mathcal{Q}, V, \delta, Q_0)$ where $\mathcal{Q}$ is a set of states, $Q_0 \subseteq \mathcal{Q}$ is the set of initial states, V is the input alphabet and δ is the transition function $\delta : \mathcal{Q} \times (V \cup \{\lambda\}) \to 2^{\mathcal{Q}}$ ($2^{\mathcal{Q}}$ is the power set of Q). The extended transition function δ^* is defined by $\delta^*(q, \lambda) = \{q\}$ and $\delta^*(q, ax) = \bigcup_{p \in S} \delta^*(p, x)$ with $S = \delta(q, a)$.

A *configuration* of $\mathbb{A}$ is a string uqv where $q \in \mathcal{Q}$ and $uv \in V^*$; the configuration

[a] "Computing by observing" is a paradigm where the computation is obtained by observing and interpreting the trajectories of a monitored system. The technique [8] was originally presented in the area of P systems developed by G. Păun and then extended to other areas [6, 7].

denotes the current state q, the read input u and the input yet to be read v. A configuration is initial when $u = \lambda$ and $q \in Q_0$. The automaton $\mathbb{A}$ can move from a configuration $C_1 = uq_1av$ to a configuration $C_2 = uaq_2v$, where $q_1, q_2 \in \mathcal{Q}$, $uv \in V^*$ and $a \in V$, if $q_2 \in \delta(q_1, a)$; such move, called *transition*, is represented by the string $C_1 : C_2$, where : is a symbol not in $V \cup \mathcal{Q}$. A *computation* of $\mathbb{A}$ on input $v \in V^*$ from initial state $q \in Q_0$ is a finite sequence of transitions (represented as $\langle qv : C_1, C_1 : C_2, \cdots, C_i : C_{i+1}, C_{i+1} : C_{i+2}, \cdots, C_{k-1} : vq' \rangle$). For a non-deterministic FSA there may be several computations on v. The set of all possible computations of $\mathbb{A}$ on $v \in V^*$ starting from the state $q \in Q_0$ is denoted by $\mathbb{A}(q, v)$; $\mathbb{A}(v) = \cup_{q \in Q_0} \mathbb{A}(q, v)$. We denote by FSA the class of finite state automata.

Following [9], an *observer* is a tuple $\mathbb{O} = (\mathcal{Q}, W, \delta, \{q\}, U, \sigma)$, where $(\mathcal{Q}, W, \delta, \{q\})$ is an FSA, with no final state and having only one initial state q; U is the output alphabet and $\sigma : \mathcal{Q} \mapsto U \cup \{\lambda\}$ is a labelling function. The *output of an observer* is the label associated to the state of the observer in which the observer halts. For a string $w \in W^*$ and an observer $\mathbb{O}$ we then write $\mathbb{O}(w)$ for this output; for a sequence $\langle w_1, \ldots, w_n \rangle$ of $n \geq 1$ strings over V^* we write $\mathbb{O}(\langle w_1, \ldots, w_n \rangle)$ for the string $\mathbb{O}(w_1) \cdots \mathbb{O}(w_n)$.

A *System/Observer system* (S/O system) is a pair constituted by an observed system $\mathbb{A} = (\mathcal{Q}, V, \delta, Q_0)$ and an observer $\mathbb{O} = (\mathcal{Q}', V \cup \mathcal{Q} \cup \{:\}, \delta', \{q'\}, U, \sigma)$. We denote such an observed system/observer by $\Omega = \mathbb{A} \oplus \mathbb{O}$. To make possible the desired interaction between the observed system and the observer, in an S/O system the input alphabet of the observer $\mathbb{O}$ must be $V \cup Q \cup \{:\}$, the alphabet used to describe transitions of the observed system $\mathbb{A}$.

In an S/O system the observer $\mathbb{O}$ translates the computations of the observed system $\mathbb{A}$ (i.e., sequences of transitions) into strings over the output alphabet of the observer. Formally, given $\Omega = \mathbb{A} \oplus \mathbb{O}$, $v \in V^*$ and $q \in Q_0$, we define $\Omega(q, v) = \{\mathbb{O}(\langle w_0, w_1, \ldots, w_n \rangle) \mid \langle w_0, w_1, \ldots, w_n \rangle \in \mathbb{A}(q, v)\}, \Omega(v) = \{\mathbb{O}(\langle w_0, w_1, \ldots, w_n \rangle) \mid \langle w_0, w_1, \ldots, w_n \rangle \in \mathbb{A}(v)\}, \Omega(V^*) = \{\mathbb{O}(\langle w_0, w_1, \ldots, w_n \rangle) \mid \langle w_0, w_1, \cdots, w_n \rangle \in \mathbb{A}(V^*)\}$. We will often (informally) refer to the strings present in the various sets (over U) Ω as *observed behaviors* of the observed system $\mathbb{A}$.

Example 1. *We construct three S/O systems. The observed system described in Figure 1 is the FSA* $\mathbb{A} = (Q, V, \delta, Q_0)$*, with* $\mathcal{Q} = \{q_0, q_1, q_2\}$*,* $V = \{a, b\}$*,* $Q_0 = \{q_0\}$*. The transition function* δ *is defined as follows:* $\delta(q_0, a) = \{q_1\}, \delta(q_0, b) = \{q_0\}, \delta(q_1, a) = \{q_1\}, \delta(q_1, b) = \{q_2\}, \delta(q_2, a) = \{q_2\}, \delta(q_2, b) = \{q_2\}$.

We consider three distinct observers $\mathbb{O}_{fin}, \mathbb{O}_{int}, \mathbb{O}_{change}$ *with output alphabet* U *and different computational powers, given by the following mappings (we fix* $p \in U$*):*

$$\mathbb{O}_{fin}(w) = \begin{cases} \lambda, & \text{if } w = zqav : zaq'v, z \in V^*, a \in V, \\ & v \in V^+, q, q' \in Q, \\ q', & \text{if } w = zqa : zaq', z \in V^*, a \in V, q, q' \in Q, \\ p \in U, & \text{if } w \notin \{zqav : zaq'v, z \in V^*, a \in V, v \in V^+, q, q' \in Q\} \\ & \cup \{zqa : zaq', z \in V^*, a \in V, q, q' \in Q\}. \end{cases}$$

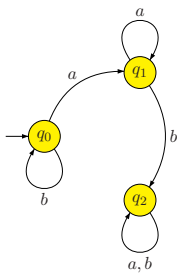


Fig. 1. The observed system is represented by the FSA $\mathbb{A} = (\{q_0, q_1, q_2, a, b, :\}, \{a, b\}, \delta, \{q_0\})$. For instance, the string $aq_1ba : abq_2a$ denotes the transition from state q_1 to state q_2 reading the symbol b. A computation of $\mathbb{A}$ is described by a sequence of strings over the alphabet of $\mathbb{A}$.

$$\mathbb{O}_{int}(w) = \begin{cases} q', & \text{if } w = zqav : zaq'v, z, v \in V^*, a \in V, q \in Q, \\ p \in U, & \text{if } w \notin \{zqav : zaq'v, z, v \in V^*, a \in V, q \in Q\}. \end{cases}$$

$$\mathbb{O}_{change}(w) = \begin{cases} c, & \text{if } w = zqav : zaq'v, z, v \in V^*, a \in V, \\ & q, q' \in Q, q \neq q', c \notin V \cup Q \cup \{\lambda\} \\ u, & \text{if } w = zqav : zaqv, z, v \in V^*, a \in V, q \in Q, \\ p \in U, & \text{if } w \notin \{zqav : zaq'v, z, v \in V^*, a \in V, q, q' \in Q, q \neq q'\} \\ & \cup \{z, v \in V^*, a \in V, q \in Q\}. \end{cases}$$

FAS implementations of the observers are described in Figures 2 and 3. We can compose the observed system $\mathbb{A}$ (Fig. 1) with the above defined observers and, for each composition, we obtain a specific observed behavior of the system $\mathbb{A}$. For instance, $\Omega_{change}(q_0, aabb) = \{cucu\}, \Omega_{fin}(q_0, aabb) = \{q_2\}, \Omega_{int}(q_0, aabb) = \{q_1q_1q_2q_2\}$:

changing the observer, we get different observed behaviors for $\mathbb{A}$, as is discussed in Figure 4.

3. Expressing Deutsch's problem in terms of FSA's

Given a Boolean function $f : \{0, 1\} \to \{0, 1\}$ and a black box for computing this function, Deutsch's problem asks to test whether f is *constant* (that is, $f(0) = f(1)$) or *balanced* ($f(0) \neq f(1)$) using only one query on the black box.

The quantum technique pioneered Deutsch in [11] "embeds" the classical computing box (given by f) into a quantum box, then use the quantum device on a "superposition" state, and finally make a single measurement of the output produced. The problem was extended in [12] and fully solved in [10] (see [5, 14, 16, 4]). The quantum solution is obtained with probability one. The de-quantisation in [4] is deterministic and relies on an efficient classical superposition; this technique works when the quantum algorithm does not use entanglement [2].

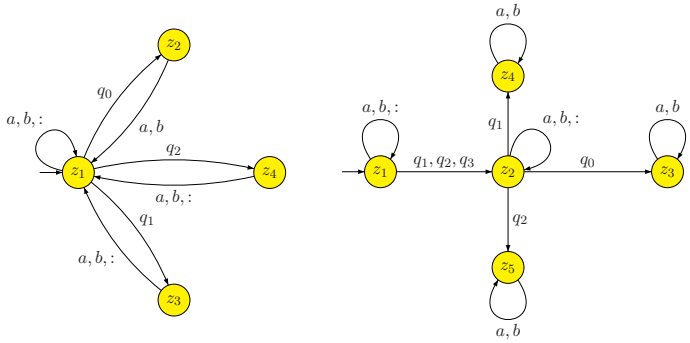


Fig. 2. The observer in the left is $\mathbb{O}_{fin} = (Z, W, \delta, \{z_1\}, U, \sigma)$, where $Z = \{z_1, z_2, z_3, z_4\}, U = \{q_0, q_1, q_2, \lambda\}$, $\sigma(z_1) = \lambda, \sigma(z_2) = q_0, \sigma(z_3) = q_1, \sigma(z_4) = q_2$; in the right, the observer is $\mathbb{O}_{int} = (Z, W, \delta, \{z_1\}, U, \sigma)$ where $Z = \{z_1, z_2, z_3, z_4, z_5\}, U = \{q_0, q_1, q_2\}$, $\sigma(z_1) = \sigma(z_2) =$ any $p \in U$, $\sigma(z_3) = q_0, \sigma(z_4) = q_1, \sigma(z_5) = q_2$. Each observer takes as input a string representing a transition of the observed system and output the symbol associated to the state where the observer stops. For example, if observer $\mathbb{O}_{fin}$ reads the string representing the transition $aq_1ba : abq_2a$ (of the observed system in Figure 1), then the observer stops in the state z_1, hence the observer output the symbol $\lambda = \sigma(z_1)$. The observer $\mathbb{O}_{fin}$ can watch the state of the observed system only when this has completely read its input while $\mathbb{O}_{int}$ can watch any state passed by the observed system processing its input.

We show that Deutsch's problem is equivalent to the problem of identifying a certain unknown FSA, in a given class of FSA's, using a specific observer. The success of such individuations is related to the computational power of the observer, and the way the observer is implemented.

In comparing the quantum solution with the classical solution proposed here it is important to note the role played by the "new black box" in which the original black box is embedded. The quantum solution embeds the classical box into a quantum black box capable of computing with superpositions, a feature unavailable to the original box. Our new black box has the capability of evaluating on strings not only on 0 and 1, again, a feature unavailable to the original box. It is a difficult open problem to define and evaluate the complexity of the embedding; see more in [2].

Let $\mathbb{A} = (\mathcal{Q}, V, \delta, Q_0)$ be an arbitrary FSA and $a \in V$ an arbitrary symbol. $\mathbb{A}$ is *a-constant* if there exists $q \in \mathcal{Q}$ such that for any $q' \in \mathcal{Q}$, $\delta(q', a) \subseteq \{q\}$. If $\mathbb{A}$ is not a-constant, then it is *a-balanced*.

Let $f_1, f_2, f_3, f_4 : \{0, 1\} \to \{0, 1\}$ be the four Boolean functions that appear in Deutsch's problem, i.e. the functions defined by $f_1(0) = 0, f_1(1) = 1$, $f_2(0) = 1, f_2(1) = 0$, $f_3(0) = 0, f_3(1) = 0$, $f_4(0) = 1, f_4(1) = 1$.

Without adding extra information, we can associate to these functions four FSA's, with states $\mathcal{Q} = \{q_0, q_1\}$:

$\mathbb{A}_1 = (\mathcal{Q}, \{a\}, \delta_1, \mathcal{Q})$ with $\delta_1(q_0, a) = \{q_0\}$ and $\delta_1(q_1, a) = \{q_1\}$, $\mathbb{A}_2 = (\mathcal{Q}, \{a\}, \delta_2, \mathcal{Q})$ with $\delta_2(q_0, a) = \{q_1\}$ and $\delta_2(q_1, a) = \{q_0\}$, $\mathbb{A}_3 = (\mathcal{Q}, \{a\}, \delta_3, \mathcal{Q})$ with

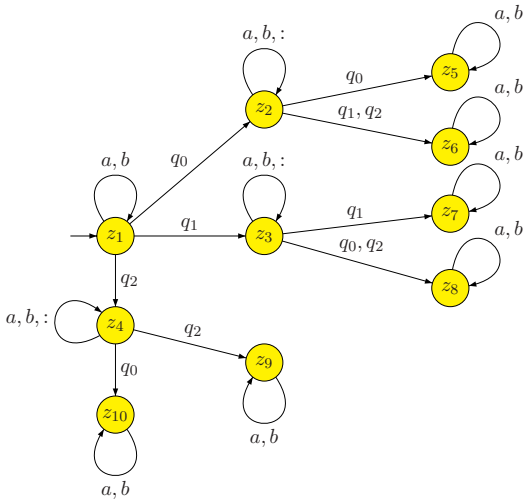


Fig. 3. The observer $\mathbb{O}_{change} = (Z, W, \delta, \{z_1\}, U, \sigma)$, where $Z = \{z_1, \ldots, z_{10}\}, W = \{a, b, q_1, q_2, q_3\}, U = \{u, c\}, \sigma(z_1) = \sigma(z_2) = \sigma(z_3) = \sigma(z_4) =$ any p in $U, \sigma(z_5) = \sigma(z_7) = \sigma(z_9) = u, \sigma(z_6) = \sigma(z_8) = \sigma(z_{10}) = c$. The observer output u or c depending on the observed transitions. For instance, the observer reading the transition $aq_1ba : abq_2a$, stops in the state z_8 and then output the symbol c. The observer $\mathbb{O}_{change}$ can see when the observed system has changed its state. If the observers presented in Figures 2 and 3 read a string that syntactically does not represent a transition then they output any p from U.

	$q_0aabb: aq_1aabb$	$aq_1aabb:aaq_1bb$	$aaq_1bb:aabq_2b$	$aabq_2b:aabbq_2$
	↓	↓	↓	↓
$\Omega_{change}(q_0,aabb)$	c	u	c	u
$\Omega_{int}(q_0,aabb)$	q_1	q_1	q_2	q_2
$\Omega_{fin}(q_0,aabb)$	λ	λ	λ	q_2

Fig. 4. The three S/O systems Ω_{change}, Ω_{fin} and Ω_{int} are obtained by coupling $\mathbb{A}$, system is the FSA $\mathbb{A}$ described in Figure 1, with observers,$\mathbb{O}_{fin}, \mathbb{O}_{int}$ and $\mathbb{O}_{change}$ described in Figures 2 and 3. The observed behaviours of $\mathbb{A}$ $\Omega_{fin}(q_0, aabb), \Omega_{int}(q_0, aabb)$ and $\Omega_{change}(q_0, aabb)$ are then presented.

$\delta_3(q_0, a) = \{q_0\}$ and $\delta_3(q_1, a) = \{q_0\}$, $\mathbb{A}_4 = (\mathcal{Q}, \{a\}, \delta_4, \mathcal{Q})$ with $\delta_4(q_0, a) = \{q_1\}$ and $\delta_4(q_1, a) = \{q_1\}$. In this way, a black box that computes one of the functions f_i can be seen as a black box simulating the corresponding FSA $\mathbb{A}_i$.

Therefore, we can reformulate *Deutsch's problem* using FSA's $\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3$ and $\mathbb{A}_4$ (see Figure 5), in the following way: *Given a black box that simulates an FSA* $\mathbb{A} \in \{\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4\}$, *decide whether or not* $\mathbb{A}$ *is a-constant or a-balanced, by using*

only one input (one query on the black box). In other words, the Deutsch's problem is equivalent with the problem of deciding, given an arbitrary unknown FSA $\mathbb{A} \in \{\mathbb{A}_1, \mathbb{A}_2, \mathbb{A}_3, \mathbb{A}_4\}$, whether $\mathbb{A}$ is a-constant or a-balanced, by providing to $\mathbb{A}$ a single input. In the paper this is referred as *(reformulated) Deutsch's problem.*

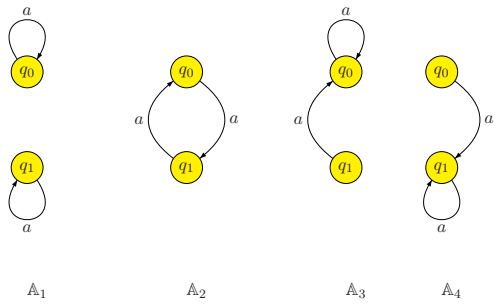


Fig. 5. Original FSA's.

The next step is to provide an *embedding* of each FSA $\mathbb{A}_i$, $i = 1, \ldots, 4$, into an FSA $\mathbb{A}_i^b$, $i = 1, \ldots, 4$ (see Figure 6), in such a way that the black box simulating $\mathbb{A}_i$ is not "opened", that is, the operation of embedding does not use/depend on any specific information identifying $\mathbb{A}_i$.

For each $\mathbb{A}_i = (\mathcal{Q}, \{a\}, \delta_i, \mathcal{Q})$, $i = 1 \ldots 4$, we define the FSA $\mathbb{A}_i^b = (\mathcal{Q}, \{a, b\}, \gamma_i, \mathcal{Q})$ with $\gamma_i(q_j, a) = \delta_i(q_j, a)$ for $j = 0, 1$, $\gamma_i(q_0, b) = \{q_1\}$ and $\gamma_i(q_1, b) = \{q_0\}$.

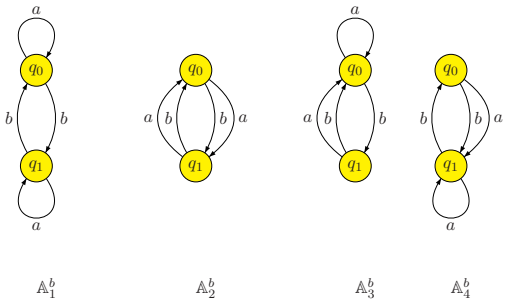


Fig. 6. "Embedded" FSA's.

Observe that $\mathbb{A}_1^b = \mathbb{A}_1 \cup \mathbb{A}$, $A_2^b = \mathbb{A}_2 \cup \mathbb{A}$, $\mathbb{A}_3^b = \mathbb{A}_3 \cup \mathbb{A}$, $A_4^b = \mathbb{A}_4 \cup \mathbb{A}$, where $\mathbb{A} = (\mathcal{Q}, \{b\}, \delta, \mathcal{Q})$ for some $b \neq a$, and $\delta(q_0, b) = \{q_1\}$ and $\delta(q_1, b) = \{q_0\}$. (We recall that, given two finite state automata $A = (Q, \Sigma, \delta, Q_0)$ and $A' = (Q', \Sigma', \delta', Q_0')$, the *union*

of A and A' is the finite state automaton $A \cup A' = (Q \cup Q', \Sigma \cup \Sigma', \delta \cup \delta', Q_0 \cup Q_0')$, with $\delta \cup \delta'(q,a) = \delta(q,a) \cup \delta'(q,a)$, for each $q \in Q \cap Q'$; $\delta \cup \delta'(q,a) = \delta(q,a)$ for each $q \in Q \setminus Q'$; $\delta \cup \delta'(q,a) = \delta'(q,a)$, for each $q \in Q \setminus Q'$.)

The proposed embedding (transforming A_i in A_i^b) is essentially a function $F : FSA \to FSA$ by $F(X) = X \cup A$. The function plays a similar role as the standard quantum embedding used by Deutsch's algorithm. The following lemma is a simple consequence of the definitions.

Lemma 2. *(i) $\mathbb{A}_i$ is a-balanced iff $\mathbb{A}_i^b$ is a-balanced, for any $i = 1 \dots 4$. (ii) $\mathbb{A}_i$ is a-constant iff $\mathbb{A}_i^b$ is a-constant, for any $i = 1 \dots 4$.*

We now consider S/O systems obtained by coupling the FSA's $\mathbb{A}_i^b$, $i = 1 \dots 4$, (as observed systems) with the observers $\mathbb{O}_{change}, \mathbb{O}_{fin}, \mathbb{O}_{int}$ defined in Example 4. We present results that show how the reformulated Deutsch's problem (i.e., deciding if an unknown observed system is a-constant or a-balanced) can be solved depending on the computational power of the allowed observer and on the possibility of finding a "smart" input for the observed system.

Consider the FSA's $\mathbb{A}_i^b = (\mathcal{Q}, \{a,b\}, \gamma_i, \mathcal{Q})$, $i = 1 \dots 4$ and let $\Omega_{fin}^i = \mathbb{A}_i^b \oplus \mathbb{O}_{fin}$, $i = 1 \dots 4$. Dividing all the possible inputs on their lengths (odd/even) or on their number of symbols bs, and assuming an arbitrary initial state for the observed system, one can prove the following result.

Theorem 3. *Given an arbitrary S/O system $\Omega = \mathbb{A} \oplus \mathbb{O}_{fin} \in \{\Omega_{fin}^1, \Omega_{fin}^2, \Omega_{fin}^3, \Omega_{fin}^4\}$ there exist no input $w \in \{a,b\}^*$ and no computable function $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$, such that $f(\Omega(w)) = \{1\}$ iff $\mathbb{A}$ is a-constant.*

Proof. We start by remarking that, for all $w \in \{a,b\}^*$, $\Omega = \mathbb{A} \oplus \mathbb{O}_{fin} \in \{\Omega_{fin}^1, \Omega_{fin}^2, \Omega_{fin}^3, \Omega_{fin}^4\}$ and any initial state of $\mathbb{A}$, there exists at least a computation on input w of $\mathbb{A}$. However, it is possible to decide whether $\mathbb{A}$ is a-constant or a-balanced using the set $\Omega_{fin}(w)$ if and only if $\Omega_{fin}(w) = \{q_i\}$ (if $\mathbb{A}$ is a-balanced) and $\Omega_{fin}(w) = \{q_j\}$ (if $\mathbb{A}$ is a-constant), where $\{i,j\} = \{1,2\}$. Consequently, there would exist an input w for which, independently of the initial (starting) state of the $\mathbb{A}_1^b, \mathbb{A}_2^b$ (the a-balanced FSA's), $\Omega_{fin}^1(w), \Omega_{fin}^2(w)$ are the set $\{q_i\}$, $i \in \{1,2\}$. However this is impossible as $\mathbb{A}_2^b$ stops in a state different than the initial state on inputs of odd length and a final state identical with the initial state on inputs of even length. Hence, there is no computation of $\mathbb{A}$ that produces the same final state for $\mathbb{A}_2^b$ no matter what is the initial state of $\mathbb{A}_2^b$. A similar argument can be found for $\mathbb{A}_1^b$ by taking into account not the parity of the length of input w, but the parity of the number of occurrences of b in w: an odd number of occurrences of b in w changes the state, an even number conserves the state. Therefore the function f cannot be constructed and this proves the theorem. □

However, the reformulated Deutsch's problem can be solved with one input if one permits the observer $\mathbb{O}_{int}$. One can check that the two inputs *aaba*, or *abaa*, can

be used to determine if the observed system is a-constant or a-balanced. Moreover, any input whose length is equal or shorter than 3, is not enough to determine the type of the observed system.

Theorem 4. *Consider the FSA's* $\mathbb{A}_i^b = (\mathcal{Q}, \{a,b\}, \gamma_i, \mathcal{Q})$ *and let* $\Omega_{int}^i = \mathbb{A}_i^b \oplus \mathbb{O}_{int}$, $i = 1 \ldots 4$. *Let* $\Omega = \mathbb{A} \oplus \mathbb{O}_{int} \in \{\Omega_{int}^1, \Omega_{int}^2, \Omega_{int}^3, \Omega_{int}^4\}$ *be an arbitrary S/O system.*

(i) There exist no input $w \in \{a,b\}^*$ *with* $|w| \leq 3$ *and no computable function* $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$ *such that* $f(\Omega(w)) = \{1\}$ *iff* $\mathbb{A}$ *is a-constant.*

(ii) There exist an input $w \in \{a,b\}^*$ *with* $|w| = 4$ *and a computable function* $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$, *such that* $f(\Omega(w)) = \{1\}$ *iff* $\mathbb{A}$ *is a-constant. Moreover, there exist a computable function* $f' : \Omega(\{a,b\}^*) \longrightarrow \{1,2,3,4\}$, *and an input* $w \in \{a,b\}^*$ *such that* $f'(\Omega(w)) = \{i\}$ *iff* $\Omega = \Omega_{int}^i$ *for any* $i = 1 \ldots 4$.

Proof. (i) We first prove that any input of length at most 3 cannot be used to decide if $\mathbb{A}$ is a-constant or a-balanced. If $|w| = 1$ the result derives as in the proof of Theorem 3. If $|w| = 2$, we have two cases:

(1) if $w = aa$ or $w = ab$, then $\Omega_{int}^1(w, q_0) = \Omega_{int}^3(w, q_0)$.
(2) if $w = ba$ or $w = bb$, then $\Omega_{int}^1(w, q_1) = \Omega_{int}^3(w, q_1)$.

If $|w| = 3$, we have four cases:

(1) if $w = aaa$ or $w = aab$, then $\Omega_{int}^1(w, q_0) = \Omega_{int}^3(w, q_0)$.
(2) if $w = aba$ or $w = abb$, then $\Omega_{int}^2(w, q_0) = \Omega_{int}^4(w, q_0)$.
(3) if $w = baa$ or $w = bab$, then $\Omega_{int}^1(w, q_1) = \Omega_{int}^3(w, q_1)$.
(4) if $w = bba$ or $w = bbb$, then $\Omega_{int}^1(w, q_1) = \Omega_{int}^4(w, q_1)$.

As we can see there is no input that can differentiate the case when $\mathbb{A}$ is a-balanced; therefore the function f cannot exist. (ii) We prove now that $w = aaba$ can precisely identify Ω, i.e., it can also decide if $\mathbb{A}$ is a-constant or a-balanced. For doing this we show that for each Ω_{int}^i, $i = 1 \ldots 4$, the result is different.

If $\Omega = \Omega_{int}^1$, the two possibilities with $aaba$ are: $\Omega(aaba, q_0) = \{q_0q_0q_1q_1\}$, $\Omega(aaba, q_1) = \{q_1q_1q_0q_0\}$. If $\Omega = \Omega_{int}^2$, the two possibilities with $aaba$ are $\Omega(aaba, q_0) = \{q_1q_0q_1q_0\}$,$\Omega(aaba, q_1) = \{q_0q_1q_0q_1\}$. If $\Omega = \Omega_{int}^3$, the two possibilities with $aaba$ are $\Omega(aaba, q_0) = \{q_0q_0q_1q_0\}$, $\Omega(aaba, q_1) = \{q_0q_0q_1q_0\}$. If $\Omega = \Omega_{int}^4$, the two possibilities with $aaba$ are $\Omega(aaba, q_0) = \{q_1q_1q_0q_1\}$, $\Omega(aaba, q_1) = \{q_1q_1q_0q_1\}$.

The computable function $f' : \Omega(\{a,b\}^*) \longrightarrow \{1,2,3,4\}$ can be then defined by:

$$f'(v) = \begin{cases} 1, & \text{if } v \in \{q_0q_0q_1q_1, q_1q_1q_0q_0\}, \\ 2, & \text{if } v \in \{q_1q_0q_1q_0, q_0q_1q_0q_1\}, \\ 3, & \text{if } v \in \{q_0q_0q_1q_0, q_0q_0q_1q_0\}, \\ 4, & \text{if } v \in \{q_1q_1q_0q_1, q_1q_1q_0q_1\}. \end{cases}$$

It is easy to verify that $f'(\Omega(w)) = \{i\}$ iff $\Omega = \Omega_{int}^i$ for any $i = 1 \ldots 4$.

We can also define $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$ by

$$f(v) = \begin{cases} 0, & \text{if } f'(v) \in \{1,2\}, \\ 1, & \text{if } f'(v) \in \{3,4\}, \end{cases}$$

and check that $f(\Omega(w)) = \{1\}$ iff $\mathbb{A}$ is a-constant.

In a similar way one can show that there are only two inputs of length 4, *aaba* and *abaa*, that can distinguish the two classes and identify Ω. □

If we decrease the "observational power" by working with the observer $\mathbb{O}_{change}$ one can still decide if the automaton $\mathbb{A}$ is a-balanced or a-constant using a specific input, but, in this case, it is not possible to identify the precise automaton. In fact, the input *aaba* (or *abaa*) can be used to differentiate if the observed system is a-constant or a-balanced. Any other input of length shorter than 4 is not enough to distinguish the type of observed system.

Theorem 5. *Consider the FSA's* $\mathbb{A}_i^b = (\mathcal{Q}, \{a,b\}, \gamma_i, \mathcal{Q})$, *with* $i = 1 \dots 4$ *and* $\Omega^i_{change} = \mathbb{A}_i^b \oplus \mathbb{O}_{change}$ *for* $i = 1 \dots 4$. *Let* $\Omega = \mathbb{A} \oplus \mathbb{O}_{change} \in \{\Omega^1_{change}, \Omega^2_{change}, \Omega^3_{change}, \Omega^4_{change}\}$ *be an arbitrary S/O system.*

(i) There exist no input $w \in \{a,b\}^*$ *with* $|w| \leq 3$ *and computable function* $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$ *such that* $f(\Omega(w)) = \{1\}$ *iff* $\mathbb{A}$ *is a-constant.*

(ii.a) There exist an input $w \in \{a,b\}^*$ *with* $|w| = 4$ *and a computable function* $f : \Omega(\{a,b\}^*) \longrightarrow \{0,1\}$ *such that* $f(\Omega(w)) = \{1\}$ *iff* $\mathbb{A}$ *is a-constant. (ii.b) Moreover, there are no computable function* $f' : \Omega(\{a,b\}^*) \longrightarrow \{1,2,3,4\}$ *and input* $w \in \{a,b\}^*$ *such that* $f'(\Omega(w)) = \{i\}$ *iff* $\Omega = \Omega^i_{change}$, $i = 1 \dots 4$.

Proof. (i) The proof is similar to that of Theorem 4. (ii.a) We show that, using the input $w = aaba$ one can differentiate $\mathbb{A} \oplus \mathbb{O}_{change}$, with $\mathbb{A}$ a-constant, from $\mathbb{A} \oplus \mathbb{O}_{change}$ with $\mathbb{A}$ a-balanced. There are only four possible cases. If $\Omega = \Omega^1_{change}$, we have $\Omega(aaba, q_0) = \{ucu\}, \Omega(aaba, q_1) = \{ucu\}$. If $\Omega = \Omega^2_{change}$, we have $\Omega(aaba, q_0) = \{ccc\}, \Omega(aaba, q_1) = \{ccc\}$. If $\Omega = \Omega^3_{change}$, we have $\Omega(aaba, q_0) = \{ucc\}, \Omega(aaba, q_1) = \{ucc\}$. If $\Omega = \Omega^4_{change}$, we have $\Omega(aaba, q_0) = \{ucc\}, \Omega(aaba, q_1) = \{ucc\}$.

It is easy to verify that $f(\Omega(w)) = \{1\}$ iff $\mathbb{A}$ is a-constant for the function $f : \Omega(V^*) \longrightarrow \{0,1\}$ defined by

$$f(v) = \begin{cases} 0, & \text{if } v \in \{ucu, ccc\}, \\ 1, & \text{if } v \in \{ucc\}. \end{cases}$$

(ii.b) The only two inputs of length 4 that can differentiate the S/O system Ω, where $\mathbb{A}$ is a-constant automaton from the S/O systems where $\mathbb{A}$ is an a-balanced automaton, are *aaba* and *abaa*. However, they cannot distinguish the S/O systems having $\mathbb{A}$ as a-balanced FSA's. Indeed, $\Omega^3_{change}(aaba) = \Omega^4_{change}(aaba)$ and $\Omega^3_{change}(abaa) = \Omega^4_{change}(abaa)$ which shows the impossibility to construct f'. □

The computable functions f defined in Theorems 4 and 5 can be implemented by a finite state transducer. In this way, a single finite state automaton can be obtained by a standard Cartesian product of the corresponding observers and transducers.

4. Conclusions

We have applied the technique of computing by observing to de-quantise Deutsch's quantum algorithm by isolating the external observer from the observed system. We have shown that the ability to solve Deutsch's problem depends on the computational power of the external observer and we have classified observers (as finite state automata) that can solve the problem.

Acknowledgment

We thank A. Abbott and the anonymous referee for constructive criticism.

References

[1] A. A. Abbott. The Deutsch-Jozsa Problem: De-quantization and Entanglement, *CDMTCS Research Report* 371, 2009, 16 pp; to appear in *Natural Computing*.
[2] A. A. Abbott, C. S. Calude. Understanding the quantum computational speed-up via de-quantisation, in S. B. Cooper, E. Kashefi, P. Panangaden (eds.). *Developments in Computational Models (DCM 2010)* EPTCS 26, 2010, pp. 1–12.
[3] Arvind. Quantum entanglement and quantum computational algorithms. *Pramana. Journal of Physics*, 56(2 & 3) (2001), 357–365.
[4] C. S. Calude. De-quantising the solution of Deutsch's problem, *International Journal of Quantum Information* 5, 4(2007), 1–7.
[5] C. S. Calude, G. Păun. *Computing with Cells and Atoms*, Taylor & Francis Publishers, London, 2001.
[6] M. Cavaliere. Computing by observing: A short survey, *Proceedings of Computability in Europe 2008, Logic and Theory of Algorithms*, LNCS 5028, Springer, 2008, 110–119.
[7] M. Cavaliere. Computing by observing, *Scholarpedia*, 5(1):9335, 2010.
[8] M. Cavaliere, P. Leupold. Evolution and observation. A new way to look at membrane systems, *Proceedings Workshop on Membrane Computing 2003*, LNCS 2933, Springer, 2004, 153–172.
[9] M. Cavaliere, P. Leupold. Evolution and observation: A non-standard way to generate formal languages, *Theoretical Computer Science*, 321, 2-3 (2004), 233–248.
[10] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca. Quantum algorithms revisited, *Proceedings of the Royal Society of London* Series A454 (1998), 339–354.
[11] D. Deutsch. Quantum theory, the Church-Turing principle, and the universal quantum computer, *Proceedings of the Royal Society of London* Series A400 (1985), 97–117.
[12] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation, *Proceedings of the Royal Society of London* Series A439 (1992), 553.
[13] A. Ekert, R. Jozsa. Quantum algorithms: Entanglement enhanced information processing, *Philosophical Transactions of the Royal Society* A 356, 1743, (1998),1769–1782.
[14] J. Gruska. *Quantum Computing*, McGraw-Hill, London, 1999.
[15] M. S. Hannachi, F. Dong, Y. Hatakeyama, and K. Hirota. On the use of fuzzy logic for inherently parallel computations. In *3rd International Symposium on Computational Intelligence and Intelligent Informatics*, 2007, 89–92.
[16] M. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2001.
[17] A. Salomaa, *Formal Languages*, Academic Press, 1987.