Abstract

In this thesis we study quantum-like representation and simulation of quantum algorithms by using classical computers. The quantum–like representation algorithm (QLRA) was introduced by A. Khrennikov (1997) to solve the “inverse Born’s rule problem”, i.e. to construct a representation of probabilistic data—measured in any context of science—and represent this data by a complex or more general\textsuperscript{1} probability amplitude which matches a generalization of Born’s rule. The outcome from QLRA matches the formula of total probability with an additional trigonometric, hyperbolic or hyper-trigonometric interference term and this is in fact a generalization of the familiar formula of interference of probabilities.

We study representation of statistical data (of any origin) by a probability amplitude in a complex algebra and a Clifford algebra (algebra of hyperbolic numbers). The statistical data is collected from measurements of two dichotomous and trichotomous observables respectively. We see that only special statistical data (satisfying a number of nonlinear constraints) have a quantum–like representation.

We also study simulations of quantum computers on classical computers. Although it can not be denied that great progress have been made in quantum technologies, it is clear that there is still a huge gap between the creation of experimental quantum computers and realization of a quantum computer that can be used in applications. Therefore the simulation of quantum computations on classical computers became an important part in the attempt to cover this gap between the theoretical mathematical formulation of quantum mechanics and the realization of quantum computers. Of course, it can not be expected that quantum algorithms would help to solve NP problems for polynomial time on classical computers. However, this is not at all the aim of classical simulation.

The second part of this thesis is devoted to adaptation of the Mathematica symbolic language to known quantum algorithms and corresponding simulations on classical computers. Concretely we represent Simon’s algorithm, Deutsch-Josza algorithm, Shor’s algorithm, Grover’s algorithm and quantum error-correcting codes in the Mathematica symbolic language. We see that the same framework can be used for all these algorithms. This framework will contain the characteristic property of the symbolic language representation of quantum computing and it will be a straightforward matter to include future algorithms in this framework.

Keywords: Born’s rule · Clifford algebra · Deutsch-Josza algorithm · Grover’s algorithm · Hyperbolic interferences · Inverse Born’s rule problem · Probabilistic data · Quantum computing

\textsuperscript{1}A Clifford algebra is used to this more general representation
· Quantum error-correcting · Quantum-like representation algorithm · Shor’s algorithm · Simon’s algorithm · Simulation of quantum algorithms
Acknowledgments

I am grateful to my supervisor professor Andrei Khrennikov for formulation of problems and introduction to the domains of quantum probability and information. I would like to thank Yaroslav Volovich who guided me into the research field, the simulation of quantum computers. I am very thankful to Guillaume Adenier and my co-supervisor Astrid Hilbert for all the discussions we have had about quantum mechanics. I am also thankful to my fellow Ph.D. students especially thank to my good friend Haidar Al-Talibi. I can not describe in words how grateful I am for Karoline Johansson helpfulness and support. I am also would like to thank my colleagues in The School of Computer Science, Physics and Mathematics in Linnaeus University especially thanks to Karl-Olof Lindahl, Marcus Nilsson, Per Nilsson, Per-Anders Svensson. Finally and foremost I would like to thank Carolina and my children Adam and Isak for their love and support.
Preface

This thesis consists of two parts representing two different studies in quantum probability and information. The first part is about quantum-like representation algorithm. This algorithm (Khrennikov [25]) provides a possibility- to represent statistical data of any origin (i.e., not only from physics, but even from, e.g., psychology or finances) in a quantum-like way, i.e., by complex probability amplitudes. These amplitudes reproduce probabilities via Born’s rule: the square of the absolute value of the amplitude gives the corresponding probability. the second part is a simulation of quantum algorithms by using classical computers. All basic algorithms are simulated on basis of the packet of programs Mathematica\(^2\). Thus the packet was adapted to quantum algorithms. Roughly speaking a quantum version of Mathematica was created. The thesis consists of five papers on the quantum-like representation algorithm and five on simulation of quantum computers.

Papers Included in the Thesis

I On consistency of the quantum-like representation algorithm.  
P. Nyman.

II On the consistency of the quantum-like representation algorithm for hyperbolic interference.  
P. Nyman.

III Representation of probabilistic data by complex probability amplitudes; the case of triple-valued observables.  
P. Nyman and I. Basieva.

IV Quantum-like representation algorithm for trichotomous observables.  
Accepted in *International Journal of Theoretical Physics*  
P. Nyman and I. Basieva.

V On hyperbolic interferences in the quantum-like representation algorithm for the case of triple–valued observables  
Preprint: Foundations of Physics  
P. Nyman.

VI Simulation of quantum algorithms on a symbolic computer.  
P. Nyman.

\(^2\)For more information about Mathematica see http://www.wolfram.com
VII  Simulation of Deutsch-Jozsa algorithm in mathematica.
    P. Nyman.

VIII Simulation of Simon’s algorithm in mathematica.
    P. Nyman

IX  A compact program code for simulations of quantum algorithms in classical computers.
    P. Nyman

X  Simulation of quantum error correcting code.
# Contents

1 Introduction
   - Introduction to Quantum-Like Representation Algorithm ........................................ 1
   - Introduction to Quantum Computing ................................................................. 2

2 Inversion of Born’s Rule
   - Quantum-Like Representation Algorithm .......................................................... 4
   - Interference Classification .................................................................................... 5
   - Probability Amplitudes ........................................................................................ 6
   - The Interference Term in Formula of Total Probability ....................................... 7
   - Unitarity of Transition Operator ........................................................................... 8
   - Clifford algebra (hyperbolic algebra) ................................................................. 9
   - Hyperbolic Hilbert space ..................................................................................... 10

3 Quantum Computing
   - Simulation on Classical Computer ...................................................................... 12
   - Symbolic Language for Simulation ................................................................... 13
   - Quantum Circuit ................................................................................................. 14

4 Summary of included papers
   - Paper I: On consistency of the quantum-like representation algorithm ............ 14
   - Paper II: On the consistency of the quantum-like representation algorithm for hyperbolic interference ............................................................... 15
   - Paper III: Representation of probabilistic data by complex probability amplitudes; the case of triple-valued observables ............................. 15
   - Paper IV: Quantum-like representation algorithm for trichotomous observables ........................................................................................................... 16
   - Paper V: Simulation of Quantum Algorithms on a Symbolic Computer ............ 16
   - Paper VI: On hyperbolic interferences in the quantum-like representation algorithm for the case of triple-valued observables ..................... 17
   - Paper VII: Simulation of Deutsch-Jozsa Algorithm in Mathematica ................ 17
   - Paper VIII: Simulation of Simon’s Algorithm in Mathematica ............................ 18
   - Paper IX: A compact program code for simulations of quantum algorithms in classical computers ................................................................. 18
   - Paper X: Simulation of Quantum Error Correcting Code ................................... 19

Bibliography ................................................................................................................ 19
I On consistency of the quantum-like representation algorithm. 23

II On the consistency of the quantum-like representation algorithm for hyperbolic interference. 27

III Representation of probabilistic data by complex probability amplitudes; the case of triple-valued observables. 31

IV Quantum-like representation algorithm for trichotomous observables. 35

V On hyperbolic interferences in the quantum-like representation algorithm for the case of triple-valued observables. 39

VI Simulation of quantum algorithms on a symbolic computer. 43

VII Simulation of Deutsch-Jozsa algorithm in mathematica. 47

VIII Simulation of Simon’s algorithm in mathematica. 51

IX A compact program code for simulations of quantum algorithms in classical computers. 55

X Simulation of quantum error correcting code. 59
1 Introduction

Historically the development of quantum theory begun in the middle of the 19th century. Gustav Kirchhoff’s introduced the Kirchhoff’s law of thermal radiation 1859 when he examined emission of black-body radiation [1]. Wilhelm Wien continued with studies on the emission of black-body radiation and concluded Wien’s displacement law. Moreover, the early 20th century was an important time in the development of our understanding of quantum physics. December 14, 1900 is seen by many as the day of the birth of quantum mechanics, on that day Max Planck gave a theoretical basis for Planck’s radiation law $E = hf$, where $h$ is Planck’s constant and $f$ is radiation frequency. Five years later Albert Einstein gave an explanation of the photoelectric effect. The Copenhagen interpretation was introduced by Niels Bohr and Werner Heisenberg after meetings and discussed in Copenhagen. They also discussed this interpretation with Paul Dirac and Erwin Schrödinger. The double-slit experiment also called Young’s experiment after Thomas Young who invented the experiment (but for incoherent light) in 19th Century. In this famous experiment coherent light is sent through a double slit and hits the screen. On the screen appears an interference pattern which show that the light has a wave property. Later this experiment was done for quantum light which combines the wave and particle properties. The former is exhibited through the interference pattern and the latter discrete dot-type detection pattern. This experiment have connection to the first part of this thesis and the reader can with advantage keep this interference experiment in mind. We will also consider further development of this experiment to three slits or in the general case arbitrary numbers of slits.

Introduction to Quantum-Like Representation Algorithm

Probability theory historically originated from the calculations of probabilities in games. The mathematician Gerolamo Cardano was a pioneer on this theory during the sixteenth century. Blaise Pascal’s corresponded with Pierre de Fermat concerning odds in games and published the important paper “Traité du triangle arithmétique” [2]. Jacques Bernoulli wrote an innovative book “Ars conjectandi” [3] which was published 1713, eight years after his death. The important Bayes’ law

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

for relations of conditional probability was introduced by Thomas Bayes and published 1763 in the paper “An Essay towards solving a Problem in the Doctrine of Chances” [4]. Bayes’ law can be rewritten to the law of total probability

$$P(A) = \sum_n P(A \mid B_n)P(B_n)$$
which we generalize by adding terms of interference. The generalized formula is essential in this thesis and the terms of interference associate the probability theory to quantum physics.

The interrelation between classical and quantum probabilistic data was discussed in numerous papers (from various points of view), see\(^1\), e.g., [10–24]. We are interested to represent probabilistic data by probability amplitude, where the probabilistic data have origin independence and the amplitude is complex, hyperbolic or a combination of these. We often refer to quantum states as quantum-like wave functions. The wave function is constructed by the quantum-like representation algorithm (QLRA). This algorithm representation of probabilistic data match Born’s rules and was introduced by A. Khrennikov, see e.g. [5–9]. A common feature will be to find a representation by using QLRA. It will be straightforward to find the wave function for dichotomous observables, the simplest situation with two observables and two possible values each. In this case we show the consistency of the algorithm. All probabilistic data can not be represented by complex amplitudes, in those cases the data is represented by hyperbolic amplitudes or a combination of these two amplitudes (hyper-complex amplitude). We show that also for the representation of the dichotomous case by hyperbolic amplitude is QLRA consistent and that no hyper-complex amplitude exist in the dichotomous case. The complexity of the problem to reconstruct the wave function increase remarkably if we add one more dimensional and consider the trichotomous case. We study this trichotomous case for hyperbolic and complex amplitude and constructed a class of probabilistic data that generate quantum like (QL)-amplitudes.

**Introduction to Quantum Computing**

Richard Feynman [27] proposed to make simulations of quantum mechanics which introduce the idea of making quantum computers, he also mentions that this is something different from the classical Turing machine. David Deutsch [29] developed the idea of quantum computing and introduced a quantum mechanical Turing machine analogue of the Turing machine. He also formulate a problem for quantum computing, which he solved by the Deutsch–Jozsa algorithm. Hence, Deutsch showed a prospect for the development of quantum computers. A few new algorithms were created after Deutsch–Jozsa algorithm.

However, one should be well aware of the following problem in quantum computing.\(^2\) Unfortunately, the number of known quantum algorithms and problems which can be solved is few. It is also proved that a classical algorithm can not be speed up by using a black box in a quantum computer,

---

\(^{1}\)The list of references is far from complete, see Khrennikov’s monographs [19,25] for a detailed list of references.

\(^{2}\)Here quantum computing is considered from the point of view of computer science and not really from the view of quantum physics. Thus we are not interested so much in physical processes which take place in quantum computers and the problems related to these processes, e.g. decoherence.
Moreover, it seems that the last few years were not signified by the creation of new interesting quantum algorithms. It might not be that this situation in quantum computing is occasional (so one might not simply say: “we need more time to create new quantum algorithms”). One could not exclude that there are some deep roots behind the insufficient algorithmic structure of quantum computing. Such roots may be created by computer science and the general theory of algorithms.

This is why it is so important to simulate quantum computing by using different classical languages (such as C [42], Maple [43], Mathematica [44], etc) and also by developing specific languages for quantum simulations such as [45,46], etc. Some of the programming languages used for simulations of quantum computers were conceived as programming languages for quantum computers and not only for simulations.

Of course, people performing simulations of quantum computing by using different symbolic languages as well as developing special languages for quantum programming dream about developing new quantum algorithms by clarifying the logic and algorithmic structure.

2 Inversion of Born’s Rule

Born’s Rule is connected to measurement of quantum states (wave function) $\psi$ and probability theory. A measurement of a quantum states of observable $a$ output some eigenvalues $\alpha$ corresponding to $a$. The probability to measure eigenvalue $\alpha_i$ is $|\hat{P}_\alpha^a\psi|^2$ where $\hat{P}_\alpha^a$ is a projection operator. Here we consider the inverse problem. Given some probabilistic data, match the data with Born’s Rule – to find the wave function. We consider the following situation. We begin with ordinary quantum formalism to derive basis equalities which later will be used as the basis of creation of quantum-like representation of data of any origin. Given two incompatible observables $a,b$ of any origin (i.e. yes or no), where $a \in X^n_a$ and $b \in X^m_b$ takes values from the sets

$$X^n_a = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \quad \text{and} \quad X^m_b = \{\beta_1, \beta_2, \ldots, \beta_m\} \quad (2.1)$$

— “spectra of observables”. From Born’s rule a measurement of observable $a$ or $b$ will only give one of eigenvalue $\alpha \in X^n_a$ or $\beta \in X^m_b$, respectively. We assume that the operators have non-degenerated spectra, i.e., $\alpha_i \neq \alpha_j$ and $\beta_i \neq \beta_j$ for $i \neq j$. Consider corresponding eigenvectors to this observables and eigenvalues:

$$\hat{a} e^a_{\alpha_i} = \alpha_i e^a_{\alpha_i}, \quad \hat{b} e^b_{\beta_i} = \beta_i e^b_{\beta_i}.\,$$

Denote by $\hat{P}_\alpha^a = |e^a_{\alpha_i}\rangle\langle e^a_{\alpha_i}|$ and $\hat{P}_\beta^b = |e^b_{\beta_i}\rangle\langle e^b_{\beta_i}|$ the one dimensional projection operators and by $P^a_\alpha$ and $P^b_\beta$ the observables represented by there projections. Consider also the projection operators

$$\hat{P}^a_{\alpha_i} = \sum_{j=1, j\neq i}^n P^a_{\alpha_j}. \quad (2.2)$$
Here the observable $P_{\alpha_i} = 1$ if the result of the $a$-measurement is $a = \alpha_i$ and $P_{\alpha_i} = 0$ if $a \neq \alpha_i$. The observables $P_{\beta_j}$ are defined in the same way. We have the following relation between events corresponding to measurements

$$[P_{\alpha_i} = 0] = \bigvee_{j=1, j \neq i} [P_{\alpha_j} = 1]. \quad (2.3)$$

By Born’s rule, the probability to measure the eigenvalue $\alpha$ is $P(a = \alpha) = |\langle \psi | e_\alpha \rangle|^2$, where $|e_\alpha\rangle$ is an orthonormal eigenvector. Here the probabilities given by the quantum mechanics (QM)-formalism are

$$p_{\beta} \equiv P_\psi(b = \beta) = ||\hat{P}_{\beta} \psi||^2 = |\langle \psi | e_\beta \rangle|^2, \quad (2.4)$$

$$p_{\alpha} \equiv P_\psi(a = \alpha) = ||\hat{P}_{\alpha} \psi||^2 = |\langle \psi | e_\alpha \rangle|^2,$$

where $\psi$ is a wave function. We also have the conditional (transition) probabilities given by the QM-formalism as

$$p_{\beta|a} \equiv P_\psi(b = \beta | a = \alpha) = ||\hat{P}_{\beta} \hat{P}_{\alpha} \psi||^2 / ||\hat{P}_{\alpha} \psi||^2 = |\langle e_\alpha | e_\beta \rangle|^2, \quad (2.5)$$

where $p_{\beta|a}$ is the probability to obtain the result $b = \beta$ under the condition that the result $a = \alpha$ has been obtained.

We remark that non-degeneration of the spectra implies that they do not depend on $\psi$. Moreover, the matrix of transition probabilities is doubly stochastic. We have the following conditions (compare this with classical probability theory) for these conditional probabilities;

$$\sum_{i=1}^{n} p_{\beta|a,\alpha_i} = 1, \quad (2.6)$$

Quantum-Like Representation Algorithm

We now turn to construction of quantum-like probability amplitudes for data of any origin. Let us consider the dichotomous case (i.e. put $n=m=2$ in (2.1)). We also assume that the matrix of transition probabilities is given by $P_{\beta|a} = (p_{\beta|a})$ and that there are also given probabilities $p_{\alpha} = P(a = \alpha)$ for $\alpha \in X_a^2$, and $p_{\beta} = P(b = \beta)$ for $\beta \in X_b^2$. Probabilistic data $C = \{p_{\alpha}, p_{\beta}\}$ is related to some experimental context (in physics preparation procedure).

Suppose that the matrix of transition probabilities $P_{\beta|a}$ is given. We reconstruct Born’s rule $p_{\beta} = |\langle \psi | e_\beta \rangle|^2$ by using formula

$$D \equiv |\lambda_{l,i} \sqrt{A} + \lambda_{l,j} \sqrt{B}|^2 = A + B + 2 \frac{\lambda_{l,i} \lambda_{l,j} + \lambda_{l,j} \lambda_{l,i}}{2} \sqrt{AB}, \quad (2.7)$$

where $\lambda_{l,k} \lambda_{l,k} = |\lambda_{l,k}|^2 = 1$. Moreover, if we consider a standard pure complex amplitude and we can put $\lambda_{l,k} \equiv e^{i \theta_{\beta l,k}}$. Thus,

$$\lambda_{l,ij} \equiv \frac{\lambda_{l,i} \lambda_{l,j} + \lambda_{l,j} \lambda_{l,i}}{2} = \frac{e^{i(\theta_{\beta l,i} - \theta_{\beta l,j})} + e^{i(\theta_{\beta l,j} - \theta_{\beta l,i})}}{2} = \cos(\theta_{\beta l,j} - \theta_{\beta l,i}). \quad (2.8)$$
Select $\psi_{\beta \alpha k} = \lambda_{l,k} \sqrt{p_{\alpha k}^a p_{\beta \alpha k}}$, $\psi_{\beta} = \psi_{\beta \alpha i} + \psi_{\beta \alpha j}$ and

$$\psi = \psi_{\beta 1} e_{\beta 1} + \psi_{\beta 2} e_{\beta 2},$$

where

$$e_{\beta 1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } e_{\beta 2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

then by (2.4), (2.7) and (2.9),

$$p_{b}^{b} = |\langle \psi | e_{b}^{b} \rangle|^2 = |\psi_{\beta \alpha i} + \psi_{\beta \alpha j}|^2 = \sum_{\alpha} p_{\alpha}^{a} p_{\beta \alpha}^{b | a} + 2 \lambda_{l,ij} \sqrt{\prod_{\alpha} p_{\alpha}^{a} p_{\beta \alpha}^{b | a}},$$

where the “coefficients of interference” are given by

$$\lambda_{l,ij} = \frac{p_{b}^{b} - \sum_{\alpha} p_{\alpha}^{a} p_{\beta \alpha}^{b | a}}{2 \sqrt{\prod_{\alpha} p_{\alpha}^{a} p_{\beta \alpha}^{b | a}}.}$$

In general these coefficients need not be bounded by 1. The coefficients of interference are called trigonometric- or hyperbolic coefficients depending on if $|\lambda_{l,ij}| \leq 1$ or if $|\lambda_{l,ij}| > 1$. We will return to the classification of interference in the next chapter. The formula (2.10) gives the quantum-like representation algorithm – QLRA. In the case QLRA produces complex amplitude we call the probabilistic data for trigonometric For any case of dichotomous observables and trigonometric probabilistic data $C$, QLRA produces the complex amplitude.

**Interference Classification**

Note that the coefficients of interference in (2.11) take values in $\mathbb{R}$. We divide this into following two cases of interference depending on the absolute value of $\lambda_{l,ij}$.

1. Let $|\lambda_{l,ij}| \leq 1$ and put $\lambda_{l,i} = e^{i\theta_1 \alpha_i}$, then by (2.8) we see that

$$\lambda_{l,ij} = \cos(\theta_1 \alpha_i - \theta_1 \alpha_j).$$

We refer to this interference as trigonometric interference.

2. Let $|\lambda_{l,ij}| > 1$ and put $\lambda_{l,i} = \epsilon_{l,i} e^{j\theta \alpha_i}$ where $j^2 = 1$ and $\epsilon_{l,i} = \lambda_{l,i}/|\lambda_{l,i}|$. Here the symbol $j$ is a generator of the Clifford algebra, which we also call Hyperbolic algebra. This algebra is defined and its properties are given in chapter “Clifford algebra”. We introduce the hyperbolic exponential function

$$e^{j \theta} = \cosh \theta + j \sinh \theta, \quad \theta \in \mathbb{R}.\quad (2.12)$$

We also use the identities

$$\cosh \theta = \frac{e^{j \theta} + e^{-j \theta}}{2} \quad \text{and} \quad \sinh \theta = \frac{e^{j \theta} - e^{-j \theta}}{2j}.\quad (2.13)$$
By (2.8) it follows that
\[ \lambda_{l,ij} = \epsilon_{l,ij} \cosh(\theta_{\beta_l \alpha_i} - \theta_{\beta_l \alpha_j}), \]
where \( \epsilon_{l,ij} = \epsilon_{l,i} \epsilon_{l,j} = \lambda_{l,ij} / |\lambda_{l,ij}|. \) We call this hyperbolic interference.

**Probability Amplitudes**

The wave function has been reconstructed from probabilistic data by QLRA for the case of two dichotomous observables. Consider a more general wave function reconstructed by QLRA for two multi-valued observables, see (2.1). Set \( \psi_{|\beta} = \langle \psi|e_{\beta}^b \rangle \) and consider the complex Hilbert space and the hyperbolic Hilbert space. Then by Born’s rule
\[ p_{\beta}^b = |\psi_{|\beta}|^2. \]  
(2.14)

We have
\[ \psi = \sum_{\beta} \psi_{|\beta} e_{\beta}^b. \]  
(2.15)

Thus these amplitudes give a possibility to reconstruct the state (\( \psi \)-function). We remark that \( \psi = \sum_{\alpha} \hat{P}_{\alpha}^a \psi \), hence
\[ \psi_{|\beta} = \sum_{\alpha} \langle \hat{P}_{\alpha}^a \psi|e_{\beta}^b \rangle. \]  
(2.16)

Each amplitude \( \psi_{|\beta} \) can be represented as the sum of subamplitudes
\[ \psi_{|\alpha} = \sum_{\beta} \psi_{\beta|\alpha} \]  
(2.17)
given by
\[ \psi_{\beta|\alpha} = \langle \hat{P}_{\alpha}^a \psi|e_{\beta}^b \rangle = \langle \psi|e_{\alpha}^a \rangle \langle e_{\alpha}^a|e_{\beta}^b \rangle. \]  
(2.18)

Hence, one can reconstruct the state \( \psi \) on the basis of amplitudes \( \psi_{\beta|\alpha} \). We remark that
\[ |\psi_{|\beta_1|\alpha_1}|^2 = |\langle \psi|e_{\alpha_1}^a \rangle \langle e_{\alpha_1}^a|e_{\beta_1}^b \rangle|^2 = p_{\alpha_1}^a p_{\beta_1|\alpha_1}^{b|a}. \]  
(2.19)

In this notations
\[ p_{\beta_1|\alpha_1}^{b|a} = |\psi_{|\beta_1|\alpha_1} + \psi_{|\beta_2|\alpha_1}|^2/(p_{\alpha_1}^a + p_{\alpha_2}^a), \]  
(2.20)

where \( p_{\beta_1|\alpha_1}^{b|a} = P_{\psi}(b = \beta_1|a = \alpha_i \lor a = \alpha_j) \) Here \( |\psi_{|\beta_1|\alpha_1}| = \sqrt{p_{\alpha_1}^a p_{\beta_1|\alpha_1}^{b|a}} \) and therefore
\[ \psi_{|\beta_1|\alpha_1} = \sqrt{p_{\alpha_1}^a p_{\beta_1|\alpha_1}^{b|a}} \lambda_{l,i}, \]  
(2.21)

where \( |\lambda_{l,i}| = 1. \) Moreover, put
\[ \langle \psi|e_{\alpha_m}^a \rangle = \sqrt{p_{\alpha_m}^a \lambda_{\alpha_m}} , \quad \langle e_{\alpha_m}^a|e_{\beta_1}^b \rangle = \sqrt{p_{\beta_1|\alpha_m}^{b|a}} \lambda_{\theta_{\beta_1|\alpha_m}}, \]  
(2.22)
where $|\lambda_{\alpha_m}| = 1$ and $|\lambda_{\beta_{l_m}}| = 1$. Hence, it follows from (2.18) and (2.21) that
\begin{equation}
\lambda_{l,i} = \lambda_{\alpha_i} \lambda_{\beta_{l}}.
\end{equation}

We have a system of equations for phases $\psi_{\beta_l \alpha_i}$ for $i, j, l \in \{1, 2, 3\}$,
\begin{equation}
|\psi_{\beta_l \alpha_i} + \psi_{\beta_l \alpha_j}|^2 = |\langle \psi | e_{\alpha_i}^a \rangle \langle e_{\alpha_i}^b | e_{\beta_l}^a \rangle + \langle \psi | e_{\alpha_j}^a \rangle \langle e_{\alpha_j}^b | e_{\beta_l}^b \rangle|^2
\end{equation}
\begin{equation}
= |\langle \psi | e_{\alpha_i}^a \rangle \langle e_{\alpha_i}^b | e_{\beta_l}^a \rangle|^2 + |\langle \psi | e_{\alpha_j}^a \rangle \langle e_{\alpha_j}^b | e_{\beta_l}^b \rangle|^2
\end{equation}
\begin{equation}
+ \langle \psi | e_{\alpha_i} \rangle \langle e_{\alpha_j}^b | e_{\beta_l}^a \rangle \langle e_{\alpha_i} | \psi \rangle + \langle \psi | e_{\alpha_j} \rangle \langle e_{\alpha_i}^a | e_{\beta_l}^b \rangle \langle e_{\alpha_j} | \psi \rangle
\end{equation}
\begin{equation}
= p_{\alpha_i}^a p_{\beta_l \alpha_i}^b + p_{\alpha_i}^a p_{\beta_l \alpha_j}^b + 2 \lambda_{l,i} \sqrt{p_{\alpha_i}^a p_{\beta_l \alpha_i}^b p_{\alpha_j}^a p_{\beta_l \alpha_j}^b}
\end{equation}

where $\lambda_{l,i}$ are given as in (2.8);
\begin{equation}
\lambda_{l,i} \equiv \frac{1}{2} \left( \lambda_{l,i} \lambda_{l,j} + \lambda_{l,j} \lambda_{l,i} \right).
\end{equation}

Thus, the coefficients of interference $\lambda_{l,ij}$ can be written by (2.20) and (2.24) as
\begin{equation}
\lambda_{l,ij} = \frac{(p_{\alpha_i}^a + p_{\alpha_j}^a) p_{\beta_l \alpha_{ij}}^b - (p_{\alpha_i}^a p_{\beta_l \alpha_i}^b + p_{\alpha_j}^a p_{\beta_l \alpha_j}^b)}{2 \sqrt{p_{\alpha_i}^a p_{\beta_l \alpha_i}^b p_{\alpha_j}^a p_{\beta_l \alpha_j}^b}}.
\end{equation}

Note that this is a generalization of the interference coefficient for the dichotomous case. To show this, let $p_{\alpha_i}^a + p_{\alpha_j}^a = 1$ and $p_{\beta_l \alpha_{ij}}^b = p_{\beta_l}^b$ in (2.26) and we will get the interference coefficient for the dichotomous case (2.25).

### The Interference Term in Formula of Total Probability

By using (2.14) and (2.17) we obtain
\begin{equation}
p_{\beta_l}^b = \left| \sum_{i}^{n} \psi_{\beta_l \alpha_i} \right|^2
\end{equation}
\begin{equation}
= \sum_{i}^{n} |\psi_{\beta_l \alpha_i}|^2 + \sum_{1 \leq i, j \leq n} \psi_{\beta_l \alpha_i} \overline{\psi_{\beta_l \alpha_j}}
\end{equation}

From (2.19) is $|\psi_{\beta_l \alpha_i}|^2 = p_{\alpha_i}^a p_{\beta_l \alpha_i}^b$, and from (2.21) and (2.25) is $\psi_{\beta_l \alpha_i} \overline{\psi_{\beta_l \alpha_j}} = \sqrt{p_{\alpha_i}^a p_{\beta_l \alpha_i}^b p_{\alpha_j}^a p_{\beta_l \alpha_j}^b \lambda_{l,i} \lambda_{l,j}}$ Finally, we obtain
\begin{equation}
p_{\beta_l}^b = \sum_{i}^{n} p_{\alpha_i}^a p_{\beta_l \alpha_i}^b + \sum_{1 \leq j < i \leq n} 2 \lambda_{l,i} \sqrt{p_{\alpha_i}^a p_{\beta_l \alpha_i}^b p_{\alpha_j}^a p_{\beta_l \alpha_j}^b}.
\end{equation}
Here, if $|\lambda_{l,i,j}| \leq 1$ for all $l,i,j$, $i \neq j$ then we call this the case of trigonometric interference and the case where $|\lambda_{l,i,j}| > 1$ for all $l,i,j$, $i \neq j$ are called the case of hyperbolic interference. All other cases are combinations of these two cases of interference and is called hyper-trigonometric interference (i.e. for some $l,i,j$, $i \neq j$, $|\lambda_{l,i,j}| \leq 1$ and for the rest $|\lambda_{l,i,j}| > 1$). Equation (2.28) is the total probability with interference terms. It can be considered, see [26], as a perturbation of the classical formula of total probability

$$p_{\beta l}^b = \sum_{i=1}^{n} p_{\alpha i}^a p_{\beta l|\alpha i}^b.$$  \hspace{1cm} (2.29)

If all coefficients of interferences $\lambda_{l,i,j} = 0$, then (2.28) coincides with (2.29).

**Unitarity of Transition Operator**

We now remark that in quantum formalism bases consisting of $\hat{a}$- and $\hat{b}$-eigenvectors are orthogonal; hence the operator of transition from one basis to another is unitarity. We can always select the $b$-basis in the standard way with canonical basis $\{e^b_\beta\}$. This system gives the $a$-basis $\{e^a_\alpha\}$, where

$$e_{\alpha i}^a = \left(\begin{array}{c}
\sqrt{p_{\beta l_{1,1}^a}} 
\sqrt{p_{\beta l_{2,1}^a}} 
\vdots 
\sqrt{p_{\beta l_{m,1}^a}} 
\end{array}\right). \hspace{1cm} (2.30)$$

In terms of matrices,

$$U = \left(\begin{array}{cccc}
\sqrt{p_{\beta l_{1,1}^a}} \lambda_{1,1} & \sqrt{p_{\beta l_{1,2}^a}} \lambda_{1,2} & \cdots & \sqrt{p_{\beta l_{1,m}^a}} \lambda_{1,m} \\
\sqrt{p_{\beta l_{2,1}^a}} \lambda_{2,1} & \sqrt{p_{\beta l_{2,2}^a}} \lambda_{2,2} & \cdots & \sqrt{p_{\beta l_{2,m}^a}} \lambda_{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
\sqrt{p_{\beta l_{m,1}^a}} \lambda_{m,1} & \sqrt{p_{\beta l_{m,2}^a}} \lambda_{m,2} & \cdots & \sqrt{p_{\beta l_{m,m}^a}} \lambda_{m,m}
\end{array}\right)$$

is unitary where $Ue_\beta = e_\alpha$. Hence, we have the system of equations

$$\sum_{m} \sqrt{p_{\beta m,\alpha i} } \sqrt{p_{\beta m,\alpha k} } \lambda_{m,i} \lambda_{m,k} = 0 \hspace{1cm} (2.31)$$

and

$$\sum_{m} p_{\beta m,\alpha i} |\lambda_{m,i}|^2 = 1, \hspace{1cm} (2.32)$$

from the condition that $U$ is unitary.

The second system of equations (2.32) will always hold by (2.6) and $|\lambda_{m,i}| = 1$, see (2.21).

The first system of equations (2.31) can be rewritten by (2.23) as

$$\lambda_{\xi_{\alpha i}} \lambda_{\xi_{\alpha k}} \sum_{m} \sqrt{p_{\beta m,\alpha i} } \sqrt{p_{\beta m,\alpha k} } \lambda_{\theta_{\beta m,\alpha i}} \lambda_{\theta_{\beta m,\alpha k}} = 0, \hspace{1cm} (2.33)$$
or
\[ \sum_m \sqrt{p_{\beta m} p_{\beta m}^*} \lambda_{\beta m \alpha_i} \bar{\lambda}_{\beta m \alpha_k} = 0, \tag{2.34} \]
where \( \lambda_{\xi \alpha_i} \neq \lambda_{\xi \alpha_k} \).
Thus (2.34) imply unitarity of \( U \).

Clifford algebra (hyperbolic algebra)

As mentioned before we will consider the complex Hilbert space but also the hyperbolic Hilbert space. Therefore let us define a Clifford algebra called hyperbolic algebra (see [25]) with the purpose to define the hyperbolic Hilbert space. The formalism for this hyperbolic algebra is similar to conventional complex numbers. This algebra contains expressions as unit circle, Euler’s formula and conjugate. Thus, let an element \( z \) belong to the hyperbolic algebra \( G \) if and only if it have following form;
\[ z = x + jy, \quad x, y \in \mathbb{R}, \]
where \( j^2 = 1 \) and addition and multiplication are defined by \( z_1 + z_2 = x_1 + x_2 + j(y_1 + y_2) \) and \( z_1 z_2 = x_1 x_2 + y_1 y_2 + j(y_1 x_2 + y_2 x_1) \) respectively. This algebra is a commutative two-dimensional algebra with two orthonormal basis \( e_0 = 1 \) and \( e_1 = j \). The hyperbolic conjugate is defined as \( \bar{z} = x - jy \) where obviously \( \bar{z} \in G \). Moreover, the square of absolute value is defined by
\[ |z|^2 = zz = x^2 - y^2 \]
and therefore will \( |z|^2 \in G \), in fact \( |z|^2 \in \mathbb{R} \). \( |z| \) is not well defined for \( z \) such that \( |z|^2 \leq 0 \). Therefore set
\[ G_+ = \{ z \in G : |z|^2 \geq 0 \} \]
and
\[ G^*_+ = \{ z \in G : |z|^2 > 0 \}. \]
Moreover define the argument \( \text{arg} \ z \) for \( z \in G^*_+ \) as
\[ \text{arg} \ z = \arctanh \frac{y}{x} = \frac{1}{2} \ln \frac{x + y}{x - y}. \]
Notice that \( x \neq 0 \), \( x - y \neq 0 \) and \( \frac{x + y}{x - y} > 0 \), since \( z \in G^*_+ \). We also define a hyperbolic exponential function
\[ e^{j \theta} = \cosh \theta + j \sin \theta, \quad \theta \in \mathbb{R}. \]
Since \( \cosh \theta \geq 0 \), elements \( z \in G^*_+ \) with \( x < 0 \) can not be represented by \( |z|e^{j \theta} \). Therefore, in order to represent all elements \( z \in G^*_+ \) put
\[ z = \epsilon |z|e^{j \theta}, \]
where $\epsilon = x/|x|$ and $\theta = \arg z$. Moreover, this is a multiplicative semigroup. Let $z_1, z_2 \in G^*_+$, so $|z_1|^2, |z_2|^2 > 0$ then we see that $z_1 \cdot z_2 \in G^*_+$ by

$$|z_1 z_2|^2 = |\epsilon_1|z_1|e^{j\theta_1} \epsilon_2 |z_2|e^{j\theta_2}|^2 = |z_1|^2 |z_2|^2 > 0.$$ 

But, when we add two elements $z_1, z_2 \in G^*_+$ it follows that there exist elements such that $z_1 + z_2 \notin G^*_+$. Let us analyze for which of the elements $z_1, z_2 \in G^*_+$, $z_1 + z_2 \notin G^*_+$, i.e. $|z_1 + z_2|^2 < 0$. It follows that

$$|z_1 + z_2|^2 = |\epsilon_1|z_1|e^{j\theta_1} + \epsilon_2 |z_2|e^{j\theta_2}|^2 = |z_1|^2 + |z_2|^2 + 2\epsilon_1 \epsilon_2 |z_1||z_2| \cosh(\theta_1 - \theta_2).$$

From (2.35) and $\cosh(\theta_1 - \theta_2) > 0$ it follows that $|z_1 + z_2|^2 > 0$ if $\epsilon_1 \epsilon_2 = 1$. Here we consider elements $z_1, z_2 \in G^*_+$ such that $|z_1|^2 + |z_2|^2 + 2\epsilon_1 \epsilon_2 |z_1||z_2| \cosh(\theta_1 - \theta_2) > 0$. Let $\epsilon_1 \epsilon_2 = -1$ then $|z_1 + z_2|^2 > 0$ if and only if

$$\arccosh \left( \frac{|z_1|^2 + |z_2|^2}{2|z_1||z_2|} \right) > |\theta_1 - \theta_2|.$$

**Hyperbolic Hilbert space**

A hyperbolic Hilbert space $H$ is a $G$-linear inner product space. Let $x, y, z \in H$ and $a, b \in G$, then consider the inner product as a map from $H \times H$ to $G$ with the following properties:

1. Conjugate symmetry: $\langle x, y \rangle$ is the conjugate to $\langle y, x \rangle$

$$\langle x, y \rangle = \overline{\langle y, x \rangle}$$

2. Linearity with respect to the first argument:

$$\langle ax + bz, y \rangle = a \langle x, y \rangle + b \langle z, y \rangle$$

3. Non-degenerate:

$$\langle x, y \rangle = 0$$

for all $y \in H$ if and only if $x = 0$

In general, the norm $\|x\| = \sqrt{\langle x, x \rangle}$ is not well defined. But we only need the square of the norm $\|x\|^2 = \langle x, x \rangle$.

**3 Quantum Computing**

In [27] Richard Feynman pointed out that it is impossible to efficiently simulate quantum mechanics on classical computers.1 Consider a system consisting of $N$ quantum particles. According to quantum formalism it is described by the tensor product $H_N = H \otimes H \otimes \cdots \otimes H$ of the $N$ copies of the state space $H$ for a single particle. It is evident that the dimension of $H_N$ grows exponentially with $N$. Feynman’s observation was the first

---

1Simulation is efficient if the execute time in polynomial
step towards the creation of quantum computers. Nowadays research in quantum computing theory is not only the simulation of quantum mechanical structures, an idea that Feynman introduced, but also the execution of quantum algorithms for solving NP problems in polynomial time.

Quantum computing combining theoretical and experimental quantum physics, mathematics, quantum information theory and computer science, and realizations of a quantum computers provides the unique possibility for researchers working in various domains of science to expand and check their theories and models. Moreover they must merge all this research together. People working in quantum foundations often consider quantum computers as a device to check such basic principles of quantum mechanics as superposition and complementarity. Moreover, the theory about quantum computing is near connected to the adequacy of the description of composite quantum systems by tensor products of Hilbert spaces, the unitarity of evolution in the absence of measurement, the von Neumann projection postulate and Born’s rule. Scientists working in quantum information theory can finally approbate basic principles and constructions (the majority of which have been elaborated long before the quantum computing theory) as well as develop mathematical models, for example, for entanglement and decoherence. Experimenters proceed quickly by creating powerful sources of quantum bits, entangled particles and the development of new quantum technologies.

At the first stage of the development of quantum computing the main attention was paid to the creation of a few algorithms which may have applications in the future, such as the clarification of foundational questions and the development of experimental technologies for toy quantum computers operating with a few quantum bits.

A number of quantum algorithms have been developed since David Deutsch presented the first algorithm [28]. This algorithm determines whether a Boolean function $f : \{0, 1\} \rightarrow \{0, 1\}$ is balanced or constant. The Deutsch-Jozsa algorithm [29] is the generalization of Deutsch’s algorithm to a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. In classical view we need to applied function at least $2^n - 1 + 1$ times to decide the functions property (balanced or constant), while the quantum algorithm only need one interaction with the oracle to decide the property.

Simon’s algorithm [32, 33] is similar to the Deutsch-Jozsa algorithm, but, instead of determine whether the Boolean function is balanced or constant, it is designed to find the period of a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$. A comparison with the well-known classical algorithm shows that the application of Simon’s algorithm [32, 33] (of course, on a quantum computer) would imply exponential speedup.

These first quantum algorithms have no direct practical applications. However, their creation played an important role in the quantum comput-

---

2 A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is constant if $f(x) = 1$ or $f(x) = 0$ for all inputs $x$ and balanced if $f(x) = 1$ for half the inputs.
ing project. It became evident that quantum computers might increase the speed of calculations tremendously.

Two algorithms with a greater potential for direct implementation to practical application are Grover’s search algorithm [29] and Shor’s factorization algorithm [34]. Grover’s algorithm is a quantum search algorithm which searches through an unsorted list with square roots less queries than the most effective classical algorithm. Shor’s algorithm factorizes integers exponentially faster than any of the classical algorithms. It has obvious applications to cryptography.

At the first stage of the quantum computing project expectations of quick progress dominated in the quantum community. However, it seems that such high expectations were not totally justified. Numerous foundational and technological problems such as the decoherence of quantum bits and the instability of quantum structures with an already sufficiently small number of registers induced doubts about the realization of quantum computers. Although it could not be denied that great progress had been made in quantum technologies, it is clear that there still is a huge gap between the creation of experimental quantum computers [35–40] and realization of a quantum computer that can be used in applications. It is also evident that difficulties increase nonlinearly with an increase in the number of registers.

Simulation on Classical Computer

Since the realization of a quantum computer will probably take at least several more years, the simulation of quantum computations on classical computers became an important part of the quantum computing theory. Of course, one could not expect that quantum algorithms would help to solve NP problems for polynomial time on classical computers. However, this is not at all the aim of classical simulation. The classical simulation of quantum computations will cover part of the gap between the theoretical mathematical formulation of quantum mechanics and the realization of quantum computers. One of the most important problems in quantum computer science is the development of new symbolic languages for quantum computing and the adaptation of existing quantum algorithms to symbolic languages.

The second part of present thesis is devoted to the adaptation known quantum algorithms to the symbolic language Mathematica and corresponding simulation on a classical computer. Concretely we represent Simon’s algorithm, Deutsch-Josza algorithm, Grover’s algorithm, Shor’s algorithm and quantum error-correcting codes in the symbolic language Mathematica.

We show that the same framework are used for all these algorithms. This

---

3 The company D-Wave Systems, Inc have announced quantum computer with a 128 qubit processor
4 Mathematica is a well-known symbolic program used in many research and development field, i.e. engineering, mathematical, physical and technical field
framework will express the characteristic property of the symbolic language representation of quantum computing. It will be a straightforward matter to include this framework in future algorithms. This thesis consists of four papers in the field of simulation of quantum algorithms on classical computers by using the Mathematica symbolic language and one paper about simulation of error correcting code. In the paper VI “Simulation of Quantum Algorithms on a symbolic computer” we introduce the basic framework and implement Shor’s algorithm in this symbolic simulation. Simulation with the Mathematica symbolic language consists of two parts: the first part defines the characteristic property of the symbolic language and the second part implements the quantum algorithms and simulates measurement.

The part devoted to symbolic representation describes the quantum gates and quantum registers. In fact, one can consider this part as the symbolic language representation of notions of quantum mechanics used in quantum computers: superposition, tensor product, entanglement and operators. By this symbolic formalization we prepare a classical computer for operations with quantum gates and states. A sequence of quantum gates generates a quantum circuit and consequently implements a quantum algorithm.

**Symbolic Language for Simulation**

In the second part of this thesis we demonstrate that the Mathematica symbolic language can be used for all known quantum algorithms. This is a consequence of the realization of all basis laws of quantum mechanics with the help of Mathematica: the superposition of quantum states, the representation of the state of a composite system in the tensor product of Hilbert spaces describing the states of the components of the system, Schödinger’s unitary evolution and the measurement process based on von Neumann’s projection postulate.

Some blocks of the program code can be shared in all algorithms. The simulation procedure will adhere to the following pattern:

(a) definition (the framework),
(b) the flow through the quantum circuit,
(c) the measurement of a specially chosen observable on the output state.

We represent the mathematical formalism of quantum mechanics which are used in quantum computing in the symbolic language Mathematica. Thus we begin with the formalization of all basic elements e.g., qubits. We also represent the quantum circuits specific to the algorithm under investigation. At the end of the symbolic language we make representation of the measurement.

Coming back to (a) we point out that in this symbolic language we have a representation corresponding to Dirac’s formalism with bra and ket vectors, see [41]. After this, superposition and tensor product, etc. are expressed in symbolic Mathematica notations. Then, by moving to (b) we express quantum gates.
To be more precise, an arbitrary one-qubit quantum state of the form
\[ \phi = c_1 |0\rangle + c_2 |1\rangle \]
will be implemented as
\[ c_1 e[0] + c_2 e[1], \]
where \( e[j] \), \( j = 0, 1 \), are symbolic Mathematica representations of the basis vectors and \( c_j, j = 1, 2 \) are complex numbers. An arbitrary one-qubit quantum operator is symbolized as
\[ U := \{ e[0] \rightarrow c_3 e[0] + c_4 e[1], e[1] \rightarrow c_5 e[0] + c_6 e[1] \}. \]
Thus:
\[ U |(c_1 e[0] + c_2 e[1]) = (c_1 c_3 + c_2 c_5) e[0] + (c_1 c_4 + c_2 c_6) e[1] \]

### Quantum Circuit

In the same way as classical algorithms are implemented on (classical) computers quantum circuits are implemented as schemes of quantum gates. Quantum gates are given by special unitary one- or two-qubit operators. In contrast to classical gates, quantum gates are reversible.

In this thesis quantum circuits will be implemented in the symbolic language and will then be used for simulation on the basis of the scheme which is standard for quantum computing – a combination of a few quantum gates. A quantum circuit can be presented as

![Quantum Circuit Diagram](image)

where \( U_2, U_3 \) is some one-qubit gate and \( U_{1,2}, U_{2,3} \) some two-qubit gate (remark that \( U_2, U_3, U_{1,2}, U_{2,3} \) can be four different gates). This quantum circuit will be implemented as
\[ U_{2,3}|(U_{1,2}|(U_3|e[0,0]))), \]
where the index of \( U \) indicates to which of the qubits the gate is applied.

### 4 Summary of included papers

**Paper I: On consistency of the quantum-like representation algorithm**

**Summary**

In this paper we consider two dichotomous observables where the output
Paper II: On the consistency of the quantum-like representation algorithm for hyperbolic interference

Summary

In this paper we continue considering two dichotomous observables where the output from QLRA is a wave function, but here the wave function have hyperbolic interference. QLRA could be given by combination of trigonometric and hyperbolic interference, but we prove that for two dichotomous observables it does not exist any case of hyper-trigonometric interference. We proceed in the same way as in Paper I and study conditioning of observables \(a\) and \(b\). The probabilistic data are selected such that the QLRA produce the hyperbolic wave function. From this data the two observables \(a\) and \(b\), reconstruct two different wave functions \(\Psi_{b|a}\) and \(\Psi_{a|b}\) in \(H_{b|a}\) and \(H_{a|b}\) respectively. We show that \(\psi_{a|b}\) and \(\psi_{b|a}\) differ only by the multiplicative factor \(e^{i(\gamma_{a}b)}\). Hence, they belong to the same equivalence class. Thus they are two representatives of the same quantum state. This result proves consistency of QLRA.

Paper III: Representation of probabilistic data by complex probability amplitudes; the case of triple-valued observables

Joint work with Irina Basieva

Summary

In paper III we study QLRA for the case of two trichotomous observables. Although the formal scheme of QLRA works for multi-valued observables of an arbitrary dimension (see [25]) the complexity increase more than might be expected. The difficulty lies in how to find the class of probabilistic data which can be transferred into QL-amplitudes. Thus, the complexity domain of application of QLRA depends very much on the dimension. Therefore, we give some examples where we select mutually unbiased bases and see that the probabilistic data can be reconstructed as a wave function.
Paper IV: Quantum-like representation algorithm for trichotomous observables

Joint work with Irina Basieva

Summary
We continue to study mutually unbiased bases for the case of two trichotomous observables. We used a larger class of probabilistic data reconstructing the wave functions. Extended numerical simulation had been performed.

Paper V: Simulation of Quantum Algorithms on a Symbolic Computer

Summary
The capability of these simulations is shown by the implementation of Shor’s algorithm, which is a well-known quantum factorization algorithm with future applications in cryptography. The realization of Shor’s algorithm on quantum computers would provide a new possibility to destroy the safety of public-key cryptos.

Before introducing the quantum Fourier transform, which plays a basic role in Shor’s algorithm, let us consider one simple example of quantum circuit represented in the symbolic language. In general, one qubit state is represented in the symbolic language as

\[ e[\psi] = \alpha e[0] + \beta e[1], \]

where \( \alpha \) and \( \beta \) are complex numbers in the way that the sum of squares of their absolute values is equal to one. Here \( e[0] \) and \( e[1] \) are representations of Dirac’s notations in Mathematica. Generalization to the case of multi-qubits is evident. Let us regard the definitions part as executed and then apply this to a state \( e[00] \) in the following quantum gate.

Consider the Pauli operator \( X = |1\rangle \langle 0| + |0\rangle \langle 1| \) and the Hadamard operator \( H = 1/\sqrt{2}(|0\rangle \langle 0| + |1\rangle \langle 1| + |0\rangle \langle 0| - |1\rangle \langle 1|) \) and the quantum circuit

![Quantum Circuit Diagram]

Let the initial state be \( e[00] \). We apply \( X_2 \), \( H_1 \) and \( H_2 \) to the first and second qubit. Thus the new state will be

\[ H_1 |H_2| X_2 |e[00] = 1/2 ( e[00] - e[01] + e[10] - e[11]) \]

where we leaving out the parentheses.

This is straightforward in simulation in Mathematica;

\[ In[10] := H_1 (H_2 (X_2 e[0,0])) \]
The result is given by
\[
\text{out}[10] := \frac{1}{2} e[0,0] - \frac{1}{2} e[0,1] + \frac{1}{2} e[1,0] - \frac{1}{2} e[1,1]
\]

Shor’s algorithm contains a part of classical algorithm and the quantum Fourier transform (QFT). The QFT is a quantum analogous to the classical discrete Fourier transform. QFT contains the following quantum gates: Hadamard operator $H$, Controlled Not operator and Rotation operator $R$. The implementation of QFT will follow from the QFT Circuit:

\[
\left| j_1 \right> \xrightarrow{H} \left| R_2 \right> \cdots \left| R_{n-1} \right> \left| R_n \right> \\
\left| j_2 \right> \\
\vdots \\
\left| j_{n-1} \right> \\
\left| j_n \right> \xrightarrow{H} \left| R_2 \right> \cdots \left| R_{n-1} \right> \\
\]

where we omit the Controlled Not gates which swap the order of the qubits at the end of the circuit. This circuit is directly implemented by the code

\[
H_1 |R_{1,2}| \cdots |R_{1,n-1}|R_{1,n}|H_2|R_{2,3}| \cdots |R_{2,n-1}| \cdots |H_{n-1}|R_{n-1,n-2}|H_n| e[\psi]
\]

in this simulation language (parentheses are necessary in Mathematica, but omitted here).

Paper VI: On hyperbolic interferences in the quantum-like representation algorithm for the case of triple-valued observables

Summary
We study the case of hyperbolic interferences for two trichotomous observables. An intermediate result in this paper is to enlarge the domain of application of QLRA outside the class reconstructed by mutually unbiased bases. This result is for the case of trigonometric interference. We define the a Clifford algebra to study the case of hyperbolic interference and give some classes of domain of application of QLRA. This paper also contains some numerical calculation for a selected part of QLRA.

Paper VII: Simulation of Deutsch-Jozsa Algorithm in Mathematica

Summary
This paper represents Deutsch-Jozsa algorithm in the Mathematica simulation language. Deutsch-Jozsa algorithm are based on the first quantum algorithm (Deutsch’s algorithm). This simple algorithm demonstrated the opportunity of quantum computing. The commission for this algorithm is easy: we need to decide whether a Boolean function is balanced or constant.
Our aim is to make a simulation of this quantum algorithm in the symbolic language. The section definitions contains one-qubit gates and a so-called quantum oracle. A quantum oracle is an operator in a black box defined as:

\[ U_f |x\rangle^{\otimes n} |y\rangle = |x\rangle^{\otimes n} |y \oplus f(x)\rangle \]

Let us give a simplified picture of this task. Bob selects a balanced or constant boolean function and Alice’s task is to effectively determine which of property bob chosen. Alice uses a quantum computer to implement the algorithm by means of the quantum circuit:

\[
\begin{align*}
|0\rangle^{\otimes n} & \quad H_1 \quad U_f \quad H_2 \quad |1\rangle \\
|1\rangle & \quad H
\end{align*}
\]

Alice prepares the initial state \(e[0, 0, 0, \ldots, 1]\) and the quantum gate according to the scheme in the circuit

\[ H_{n-1}|\ldots H_2|H_1|U_f|H_n|\ldots H_2|H_1| e[0, 0, 0, \ldots, 1] \]

Finally, the measurement is performed.

She should apply the oracle only once, classically she need to call the function at least \(2^{n-1} + 1\) times.

**Paper VIII: Simulation of Simon’s Algorithm in Mathematica**

**Summary**
The simulation of Simon’s algorithm is performed in the third paper. The structure of this algorithm is similar to that of Deutsch-Jozsa algorithm. It also contains an oracle. The essential difference is the function given by the oracle. Bob selects a periodic function and Alice’s task is to find the period. Alice prepares the state \(e[0, 0, \ldots]\), applies the quantum gates \(e[0, 0, \ldots]\), and then apply the Hadamard gate to all qubits. After this, we repeat the Grover iteration and perform
the measurement. This flow for Grover’s algorithm is given by the following circuit:

Grover’s algorithm circuit using one Grover iteration

Paper X: Simulation of Quantum Error Correcting Code

Summary
This study considers implementations of error correction in a simulation language on a classical computer. Error correction will be necessary in quantum computing and quantum information. We will implement Shor code as an example of error corrections code. The Shor code is a development of the classical error correcting code known as majority voting. There are some great differences between quantum and classical error correcting. Measurements destroy the quantum states and another problem in quantum computing is continuous errors. Moreover, it is impossible to clone an arbitrary quantum state. In classical computing will errors imply that bits flips, but continuous errors in the quantum computing imply that the states phases flips or that the qubits flips or some combination of this errors. Shor code will overcome these problems in quantum computing.

Bibliography


Paper I

On consistency of the quantum-like representation algorithm.

Peter Nyman
On Consistency of the Quantum-Like Representation Algorithm

Peter Nyman

Received: 12 May 2009 / Accepted: 14 October 2009 / Published online: 4 November 2009
© Springer Science+Business Media, LLC 2009

Abstract In this paper we continue to study so-called “inverse Born’s rule problem”: to construct a representation of probabilistic data of any origin by a complex probability amplitude which matches Born’s rule. The corresponding algorithm—quantum-like representation algorithm (QLRA)—was recently proposed by A. Khrennikov (Found. Phys. 35(10):1655–1693, 2005; Physica E 29:226–236, 2005; Dokl. Akad. Nauk 404(1):33–36, 2005; J. Math. Phys. 46(6):062111–062124, 2005; Europhys. Lett. 69(5):678–684, 2005). Formally QLRA depends on the order of conditioning. For two observables (of any origin, e.g., physical or biological) $a$ and $b$, $b|a$- and $a|b$ conditional probabilities produce two representations, say in Hilbert spaces $H^b|a$ and $H^a|b$. In this paper we prove that under “natural assumptions” (which hold, e.g., for quantum observables represented by operators with nondegenerate spectra) these two representations are unitary equivalent. This result proves the consistency of QLRA.

Keywords Quantum-like representation algorithm · Inverse Born’s rule problem · Order of conditioning · Unitary equivalence of representations

1 Introduction

During the last 80 years tremendous efforts have been made to clarify the inter-relation between classical and quantum probabilities; see, e.g., von Neumann [1] for the first detailed presentation of this problem and see, e.g., Gudder [2–4], Svozil [5, 6], Fine [7], Garola et al. [8–10], Dvurecenskij and Pulmanova [12], Ballentine [11], O. Nánásiová et al. [13, 14], Allahverdyan et al. [15] for modern studies.1 We remark that during the last

---

1The list of references is far from complete, see Khrennikov’s monograph [25] for the detailed list of references.
30 years the main interest was attracted to Bell’s inequality; see, e.g. [16–19], for detailed a presentation. However, the basic rule of QM is Born’s rule. Therefore the study of its origin is not less (and maybe even more) important than investigations of Bell’s inequality. In this paper we continue to study so-called “inverse Born’s rule problem” as it was formulated by Khrennikov [20–24]:

**IBP (Inverse Born Problem)** To construct a representation of probabilistic data by a complex probability amplitude which matches Born’s rule.

The solution of IBP provides a possibility to represent probabilistic data by “wave functions” and operate with this data by using linear algebra (as we do in conventional QM). In a special case (for a pair of dichotomous observables) this problem was solved in [20–24] with the help of so-called quantum-like representation algorithm—QLRA.

Formally, the output of QLRA depends on the order of conditioning. For two observables \( a \) and \( b \), \( b|a \) - and \( a|b \) conditional probabilities produce two representations, say in Hilbert spaces \( H^{b|a} \) and \( H^{a|b} \). In this paper we prove that under natural assumptions these two representations are unitary equivalent. This result proves the consistency of QLRA.

“Natural assumptions” about probabilities which are used in this paper hold, e.g., for probabilistic data obtained in measurements of *quantum observers with nondegenerate spectra*; see Appendix.

We want to discuss IBP in detail. Consider probabilistic data collected in measurements of a pair of observables, say \( a \) and \( b \). These observables need not be quantum observables. They can describe measurement done in any domain of science. We have implemented a representation of such data by a complex amplitude or (in the abstract Hilbert space formalism) by the normalized vector in a complex Hilbert space. It provides us with a possibility of applying linear algebra to an operation with probabilities (through an operation with probability amplitudes). Clearly, such representation of data is not always possible to find. We should find constraints for constructing a representation of these data in a complex Hilbert space. We show that under natural assumptions one can introduce an algorithmic procedure of representation of probabilistic data by complex probability amplitudes matching Born’s rule. In quantum physics our algorithm, QLRA, just reproduces the wave function of a quantum system (in special cases of observables represented by self-adjoint operators with nondegenerate spectra; see Appendix); so, it is a quantum state reconstruction algorithm. However, QLRA can be applied for data collected in any domain of science; see, e.g., [26–28], to applications to cognitive science and psychology.

## 2 Inversion of Born’s Rule

We consider the simplest situation. There are given two dichotomous observables of any origin: \( a = \alpha_1, \alpha_2 \) and \( b = \beta_1, \beta_2 \). We set \( X_a = \{ \alpha_1, \alpha_2 \} \) and \( X_b = \{ \beta_1, \beta_2 \} \)—“spectra of observables”.

We assume that there is given the matrix of transition probabilities \( P^{b|a} = (p_{\beta|\alpha}^{b|a}) \), where \( p_{\beta|\alpha}^{b|a} \equiv P( b = \beta | a = \alpha ) \) is the probability to obtain the result \( b = \beta \) under the condition that the result \( a = \alpha \) has been obtained. There are also given probabilities \( p_{\alpha|a}^a \equiv P( a = \alpha ), \alpha \in X_a, \) and \( p_{\beta|b}^b \equiv P( b = \beta ), \beta \in X_b \). Probabilistic data \( C = \{ p_{\alpha|a}^a, p_{\beta|b}^b \} \) are related to some experimental context (in the physics preparation procedure).
IBP is to represent this data by a probability amplitude $\psi$ (in the simplest case it is complex-valued) such that Born’s rule holds for both observables:

$$p_\beta^b = |\langle \psi, e_\beta^b \rangle|^2, \quad p_\alpha^a = |\langle \psi, e_\alpha^a \rangle|^2,$$

where $\{e_\beta^b\}_{\beta \in X_b}$ and $\{e_\alpha^a\}_{\alpha \in X_a}$ are orthonormal bases for observables $b$ and $a$, respectively (so the observables are diagonal in the respective bases).

In [20–24] the solution of IBP is given in the form of an algorithm which constructs a probability amplitude from the data. Formally, the output of this algorithm depends on the order of conditioning. By starting with the matrix of transition probabilities $P_{a|b}$, instead of $P_{b|a}$, we construct another probability amplitude $\psi_{a|b}$ (the amplitude in (1) should be denoted by $\psi_{b|a}$ and other bases, $\{e_\beta^b\}_{\beta \in X_b}$ and $\{e_\alpha^a\}_{\alpha \in X_a}$. We shall see that under natural assumptions these two representations are unitary equivalent.

3 QLRA

3.1 $H^{b|a}$-Conditioning

Suppose that the matrix of transition probabilities $P^{b|a}$ is given. In [20–24] the following formula for the interference of probabilities (generalizing the classical formula of total probability) was derived: $p_\beta^b = \sum_\alpha p_\alpha^a p^{b|a}_{\beta\alpha} + 2\lambda_\beta \sqrt{\prod_\alpha p_\alpha^a p^{b|a}_{\beta\alpha}}$, where the “coefficient of interference” is given by

$$\lambda_\beta = \frac{p_\beta^b - \sum_\alpha p_\alpha^a p^{b|a}_{\beta\alpha}}{2\sqrt{\prod_\alpha p_\alpha^a p^{b|a}_{\beta\alpha}}}.$$

We shall proceed under the conditions:

1. $P^{b|a}$ is doubly stochastic.
2. Probabilistic data $C = \{p_\alpha^a, p_\beta^b\}$ consist of strictly positive probabilities.
3. The coefficients of interference $\lambda_\beta, \beta \in X_b$, are bounded by one: $|\lambda_\beta| \leq 1$.

Probabilistic data $C$ such that (3) holds are called trigonometric, because in this case we have the conventional formula of trigonometric interference: $p_\beta^b = \sum_\alpha p_\alpha^a p^{b|a}_{\beta\alpha} + 2\cos \theta_\beta \sqrt{\prod_\alpha p_\alpha^a p^{b|a}_{\beta\alpha}}$, where

$$\lambda_\beta = \cos \theta_\beta.$$

By using the elementary formula: $D = A + B + 2\sqrt{AB} \cos \theta = |\sqrt{A} + e^{i\theta} \sqrt{B}|^2$, for real numbers $A, B > 0, \theta \in [0, 2\pi]$, we can represent the probability $p_\beta^b$ as the square of the complex amplitude (Born’s rule): $p_\beta^b = |\psi_{b|a}\beta|^2$. Here

$$\psi_{b|a}\beta = \sqrt{p_{a1}^{b|a} p_{b\alpha1}^{b|a} + e^{i\theta_\beta} \prod_{\alpha=2}^{n} p_{a\alpha}^{b|a} p_{b\alpha2}^{b|a}}, \quad \beta \in X_b.$$

---

This formula can be easily derived in the conventional QM formalism; see, e.g., [25], by transition from the basis of eigenvectors for the $a$-observable to the basis of eigenvectors for the $b$-observables. We recall that in QM observables are given by self-adjoint operators. However, we proceed in the opposite direction. We would like to produce a complex probability amplitude and operator representation of the observables by using this formula.
The formula (4) gives the quantum-like representation algorithm—QLRA. For any trigonometric probabilistic data $C$, QLRA produces the complex amplitude $\psi^{b|a}$ (the normalized vector in the two dimensional complex Hilbert space, say $H^{b|a}$):

$$\psi^{b|a} = \psi^{b|a}_{\beta_1} e^{b|a}_{\beta_1} + \psi^{b|a}_{\beta_2} e^{b|a}_{\beta_2},$$

(5)

where

$$e^{b|a}_{\beta_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad e^{b|a}_{\beta_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

To solve IBP completely, we would like to have Born’s rule not only for the $b$-variable, but also for the $a$-variable: $p^{a}_{\alpha} = |\langle \psi^{b|a}, e^{a|b}_{\alpha} \rangle|^2, \alpha \in X_a$. Here the $a$-basis in the Hilbert space $H^{b|a}$ is given, see [20–24] for details, by

$$e^{a|b}_{\alpha_1} = \begin{pmatrix} \sqrt{p^{a}_{\alpha_1 \beta_1}} \\ \sqrt{p^{a}_{\alpha_1 \beta_2}} \end{pmatrix}, \quad e^{a|b}_{\alpha_2} = \begin{pmatrix} \sqrt{p^{a}_{\alpha_2 \beta_1}} \\ -\sqrt{p^{a}_{\alpha_2 \beta_2}} \end{pmatrix}.$$

It is orthonormal, since $P^{b|a}$ is assumed to be doubly stochastic. In this basis the amplitude $\psi^{b|a}$ is represented as

$$\psi^{b|a} = \sqrt{p^{b}_{\beta_1}} e^{a|b}_{\beta_1} + e^{i\theta_{\beta_1}} \sqrt{p^{b}_{\beta_2}} e^{a|b}_{\beta_2}.$$  

(6)

We recall that in QM a pure state $\Psi$ is defined as an equivalent class with respect to multipliers of the form $c = e^{i\gamma}$. We shall use a similar terminology. Each complex amplitude $\psi^{b|a}$ produced by QLRA determines a quantum-like state (representing given probabilistic data)—the equivalence class $\Psi^{b|a}$ being determined by the representative $\psi^{b|a}$.

3.2 $H^{a|b}$-Conditioning

Here

$$\psi^{a|b}_\alpha = \sqrt{p^{b}_{\beta_1}} p^{a|b}_{\alpha \beta_1} + e^{i\theta_\alpha} \sqrt{p^{b}_{\beta_2}} p^{a|b}_{\alpha \beta_2}, \quad \alpha \in X_a.$$  

(7)

For any trigonometric probabilistic data $C$, QLRA produces the complex amplitude $\psi^{a|b}$ (the normalized vector in the two dimensional complex Hilbert space, say $H^{a|b}$):

$$\psi^{a|b} = \psi^{a|b}_{\alpha_1} e^{a|b}_{\alpha_1} + \psi^{a|b}_{\alpha_2} e^{a|b}_{\alpha_2},$$

(8)

where

$$e^{a|b}_{\alpha_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad e^{a|b}_{\alpha_2} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Here the $b$-basis in the Hilbert space $H^{a|b}$ is given by

$$e^{a|b}_{\beta_1} = \begin{pmatrix} \sqrt{p^{a|b}_{\alpha_1 \beta_1}} \\ \sqrt{p^{a|b}_{\alpha_2 \beta_1}} \end{pmatrix}, \quad e^{a|b}_{\beta_2} = \begin{pmatrix} \sqrt{p^{a|b}_{\alpha_1 \beta_2}} \\ -\sqrt{p^{a|b}_{\alpha_2 \beta_2}} \end{pmatrix}.$$

In this basis the amplitude $\psi^{a|b}$ is represented as

$$\psi^{a|b} = \sqrt{p^{b}_{\beta_1}} e^{a|b}_{\beta_1} + e^{i\theta_{\beta_1}} \sqrt{p^{b}_{\beta_2}} e^{a|b}_{\beta_2}.$$  

(9)
As in the case of $H^{b|a}$-representation, the quantum-like state (representing given probabilistic data) is defined as the equivalence class $\Psi^{a|b}$ with the representative $\psi^{a|b}$.

4 Unitary Equivalence of $b|a$- and $a|b$-Representations

Thus, as we have seen by selecting two types of conditioning, we represented the probabilistic data $C = \{p_a^\alpha, p_b^\beta\}$ by two quantum-like states, $\Psi^{b|a}$ and $\Psi^{a|b}$. We are interested in the consistency of these representations.

We remark that any linear operator $W : H^{b|a} \to H^{a|b}$ induces the map of equivalence classes of the unit spheres with respect to multipliers $c = e^{i\gamma}$. We define the unitary operator $U^{a|b}_{b|a} : H^{b|a} \to H^{a|b}$ by $U^{b|a}_a(e^{b|a}_\alpha) = e^{a|b}_\alpha$, $\alpha \in X_a$. It induces the mentioned map of equivalent classes.

**Theorem** The operator $U^{a|b}_{b|a}$ maps $\Psi^{b|a}$ into $\Psi^{a|b}$ if and only if the following inter-relation of symmetry takes place for the matrices of transition probabilities $P^{b|a}$ and $P^{a|b}$:

$$p_{b\alpha}^{a|b} = p_{a\alpha}^{a|b},$$

for all $\alpha$ and $\beta$ from the spectra of observables $a$ and $b$.

**Proof** Take the representative of $\Psi^{b|a}$ given by (6). Then

$$U^{a|b}_{b|a}\Psi^{b|a} = \sqrt{p_{a1}^{b}} e^{a|b}_{a1} + e^{b\beta_1} \sqrt{p_{a2}^{b}} e^{a|b}_{a2}. \tag{11}$$

Our aim is to show that this vector is equivalent to the vector $\psi^{a|b}$ given by (8). By using $H^{b|a}$ analogs of (2) and (3) for the coefficients of interference and its cos-expression we determine $\cos \theta_{\alpha_1}$:

$$p_{a1}^a = p_{1\beta_1}^b p_{a1}^{a|b} + p_{2\beta_2}^b p_{a1}^{a|b} + 2 \cos \theta_{\alpha_1} \sqrt{p_{1\beta_1}^b p_{a1}^{a|b} p_{2\beta_2}^b p_{a1}^{a|b}} \tag{12}$$

$$\iff \cos \theta_{\alpha_1} = \frac{p_{a1}^a - p_{1\beta_1}^b p_{a1}^{a|b} - p_{2\beta_2}^b p_{a1}^{a|b}}{2 \sqrt{p_{1\beta_1}^b p_{a1}^{a|b} p_{2\beta_2}^b p_{a1}^{a|b}}}. \tag{12}$$

We also calculate

$$\psi^{a|b}_{a2} \psi^{a|b}_{a1} = p_{1\beta_1}^b \sqrt{p_{a1}^{a|b} p_{a2}^{a|b}} - p_{2\beta_2}^b \sqrt{p_{a2}^{a|b} p_{a1}^{a|b}}$$

$$- (\cos \theta_{\alpha_1} + i \sin \theta_{\alpha_1}) \sqrt{p_{1\beta_1}^b p_{2\beta_2}^b p_{a2}^{a|b} p_{a1}^{a|b}}$$

$$+ (\cos \theta_{\alpha_1} - i \sin \theta_{\alpha_1}) \sqrt{p_{1\beta_1}^b p_{2\beta_2}^b p_{a1}^{a|b} p_{a2}^{a|b}}. \tag{13}$$

where $\psi^{a|b}_{a2} = \sqrt{p_{1\beta_1}^b p_{2\beta_2}^b} e^{i\theta_{a1}}$ is given by (9). We use that $|\langle \psi^{a|b}_{a2} \rangle^2 = p_{a1}^a \iff \psi^{a|b}_{a2} = \sqrt{p_{a1}^a} (\cos \gamma_{a_2} + i \sin \gamma_{a_2})$ where $\gamma_{a_j} = \arg \psi^{a|b}_{a_j}, j \in \{1, 2\}$ and this gives that

$$\psi^{a|b}_{a2} \psi^{a|b}_{a1} = \sqrt{p_{a1}^a p_{a2}^a} (\cos (\gamma_{a2} - \gamma_{a_1}) + i \sin (\gamma_{a2} - \gamma_{a_1})). \tag{14}$$

The real part of (13) and (14) gives
\[
\sqrt{p_{a_1}^a p_{a_2}^a \cos(\gamma_{a_2} - \gamma_{a_1})} \\
= p_{b_1}^b \sqrt{p_{a_1 b_1}^{a|b} p_{a_2 b_1}^{a|b}} - p_{b_2}^b \sqrt{p_{a_2 b_2}^{a|b} p_{a_1 b_2}^{a|b}} \\
- \cos \theta_{a_1} \left( \sqrt{p_{b_1}^b p_{a_2 b_2}^{a|b} p_{a_1 b_1}^{a|b}} + \sqrt{p_{b_1}^b p_{a_2 b_1}^{a|b} p_{a_1 b_2}^{a|b}} \right).
\]  

(15)

Moreover, since \(p_{b_2}^b = 1 - p_{b_1}^b\) and from the condition that \(P^{a|b}\) is double stochastic i.e. \(p_{a_1 b_1}^{a|b} = p_{a_2 b_2}^{a|b} = 1 - p_{a_1 b_1}^{a|b} = 1 - p_{a_2 b_2}^{a|b}\), we rewrite (15)

\[
\sqrt{p_{a_1}^a p_{a_2}^a \cos(\gamma_{a_2} - \gamma_{a_1})} = (2p_{b_1}^b - 1) \sqrt{p_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b})} \\
+ \cos \theta_{a_1} (1 - 2p_{a_1 b_1}^{a|b}) \sqrt{(1 - p_{b_1}^b p_{a_1 b_1}^{a|b})}. 
\]  

(16)

Then by (2) and (3) we obtain \(\cos \theta_{b_1}\):

\[
\cos \theta_{b_1} = \frac{p_{b_1}^b - p_{a_1 b_1}^{b|a} p_{a_2 b_2}^{b|a}}{2 \sqrt{p_{a_1}^a p_{a_2}^a P_{b_1}^{b|a} P_{b_2}^{b|a}}}. 
\]  

(17)

Multiply (17) with \(2 \sqrt{p_{a_1}^a p_{a_2}^a}\) and use again that \(p_{b_2}^b = 1 - p_{b_1}^b\) and \(P^{a|b}\) is double stochastic and

\[
2 \sqrt{p_{a_1}^a p_{a_2}^a} \cos \theta_{b_1} = p_{a_1}^a - 1 + p_{b_1}^b + p_{a_2 b_1}^{b|a} - 2p_{a_1 b_1}^{b|a} p_{a_1 b_1}^{a|b}. 
\]  

(18)

We will show that \(\cos(\gamma_{a_2} - \gamma_{a_1}) = \cos \theta_{b_1}\) or equivalent, we show that

\[
2 \sqrt{p_{a_1}^a p_{a_2}^a \cos(\gamma_{a_2} - \gamma_{a_1})} = 2 \sqrt{p_{a_1}^a p_{a_2}^a \cos \theta_{b_1}}. 
\]  

(19)

Multiply \(\sqrt{p_{a_1}^a p_{a_2}^a \cos(\gamma_{a_2} - \gamma_{a_1})}\) by \(2 \sqrt{p_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b})}\) on the left-hand side (16) such that \(LHS = 2 \sqrt{p_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b})} \sqrt{p_{a_1}^a p_{a_2}^a \cos(\gamma_{a_2} - \gamma_{a_1})}\) and replace \(\cos \theta_{a_1}\) with

\[
\frac{p_{a_1}^a - p_{b_1}^b p_{a_1 b_1}^{a|b} - (1 - p_{a_1 b_1}^{a|b})(1 - p_{a_1 b_1}^{a|b})}{2 \sqrt{p_{a_1 b_1}^{a|b} p_{a_1 b_1}^{a|b} p_{a_1 b_1}^{a|b}}} 
\]  

on right-hand side

\[
LHS = 2(2p_{b_1}^b - 1) p_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b}) \\
+ (p_{a_1}^a - p_{b_1}^b p_{a_1 b_1}^{a|b} - (1 - p_{a_1 b_1}^{a|b})(1 - p_{a_1 b_1}^{a|b}))(1 - 2p_{a_1 b_1}^{a|b}) \\
= 2(2p_{b_1}^b - 1) P_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b}) \\
+ (p_{a_1}^a - 1 + p_{b_1}^b + p_{a_1 b_1}^{a|b} - 2p_{b_1}^b p_{a_1 b_1}^{a|b})(1 - 2p_{a_1 b_1}^{a|b}) \\
= 2(2p_{b_1}^b - 1) P_{a_1 b_1}^{a|b} (1 - p_{a_1 b_1}^{a|b}) \\
+ (p_{a_1}^a - 1 + p_{b_1}^b + p_{a_1 b_1}^{a|b} - 2p_{a_1 b_1}^{a|b} - 2p_{b_1}^b P_{a_1 b_1}^{a|b}) \\
- 2p_{a_1 b_1}^{a|b} (-1 + p_{b_1}^b + p_{a_1 b_1}^{a|b} - 2p_{b_1}^b P_{a_1 b_1}^{a|b}) - 2p_{b_1}^b P_{a_1 b_1}^{a|b} 
\]
\[ = 2(2p_b^b - 1)p_{a1\beta_1}^{a|b}(1 - p_{a1\beta_1}^{a|b}) \]
\[ + (p_{a1}^{a} - 1 + p_{\beta_1}^{b} + p_{a1\beta_1}^{a|b} - 2p_{a1\beta_1}^{a|b}p_{a1}^{a}) \]
\[ - 2p_{a1\beta_1}^{a}(1 - 2p_{\beta_1}^{b} + p_{a1\beta_1}^{a|b} - 2p_{\beta_1}^{b}p_{a1\beta_1}^{a|b}) \]
\[ = p_{a1}^{a} - 1 + p_{\beta_1}^{b} + p_{a1\beta_1}^{a|b} - 2p_{a1\beta_1}^{a|b}p_{a1}^{a}. \] (20)

From (18) and (20) it will follow that
\[ \frac{p_{a1}^{a} - 1 + p_{\beta_1}^{b} + p_{a1\beta_1}^{a|b} - 2p_{a1\beta_1}^{a|b}p_{a1}^{a}}{\sqrt{p_{a1\beta_1}^{b|a}p_{a1\beta_1}^{b|a}}} = \frac{p_{a1}^{a} - 1 + p_{\beta_1}^{b} + p_{a1\beta_1}^{a|b} - 2p_{a1\beta_1}^{a|b}p_{a1}^{a}}{\sqrt{p_{a1\beta_1}^{b|a}p_{a1\beta_1}^{b|a}}} \]
\[ \Leftrightarrow \]
\[ p_{a1\beta_1}^{b|a} = p_{a1\beta_1}^{a|b}. \] (21)

Therefore we will conclude that cos(\(\gamma_{a2} - \gamma_{a1}\)) = cos \(\theta_{\beta_1}\) iff \(P_{b|a}^{b} = P_{a|b}^{a}\). Let
\[ U_{b|a}^{a|b} = \left( \begin{array}{cc} \sqrt{p_{a1\beta_1}^{b|a}} & \sqrt{p_{b1\beta_1}^{b|a}} \\ \sqrt{p_{b2\beta_1}^{b|a}} & -\sqrt{p_{b2\beta_1}^{b|a}} \end{array} \right). \] (22)

Then let us show that this vector is equivalent to the vector \(\psi_{a1|b}\) given by (8)
\[ U_{b|a}^{a|b}\psi_{b|a}^{b|a} = \sqrt{p_{a1}^{a}}e_{a1}^{a|b} + e^{i\theta_{\beta_1}}\sqrt{p_{a2}^{a}}e_{a2}^{a|b} \]
\[ = \sqrt{p_{a1}^{a}}e_{a1}^{a|b} + e^{i(\gamma_{a2} - \gamma_{a1})}\sqrt{p_{a2}^{a}}e_{a2}^{a|b}. \] (23)

Then put \(\psi_{a|j}^{a|b} = \sqrt{p_{a1}^{a}}e^{i(\gamma_{a1})}\), \(j \in \{1, 2\}\) into (8)
\[ \psi_{a|j}^{a|b} = \sqrt{p_{a1}^{a}}e^{i(\gamma_{a1})}e_{a1}^{a|b} + \sqrt{p_{a2}^{a}}e^{i(\gamma_{a2})}e_{a2}^{a|b} \]
\[ = e^{i(\gamma_{a1})}U_{b|a}^{a|b}\psi_{b|a}^{b|a}. \] (24)

The complex amplitudes \(\psi_{a|b}^{a|b}\) and \(U_{b|a}^{a|b}\psi_{b|a}^{b|a}\) differ only by the multiplicative factor \(e^{i(\gamma_{a1})}\). Hence, they belong to the same equivalent class of vectors on the unit sphere. Thus they are two representatives of the same quantum state \(\Psi_{b|a}^{a}\). \(\square\)

**Appendix: The Inter-Relation of Symmetry for Matrices of Transition Probabilities in the Case of Observables with Nondegenerate Spectra**

We start with the general case of observables, which can have degenerate spectra. Let \(Q_{\beta}\) and \(P_{a}\) be two orthogonal projection operators, then by the projection postulate of QM (or simply by definition of quantum conditional probability):
\[ P_{a\beta}^{a|b} = \frac{\langle \psi | Q_{\beta} P_{a} Q_{\beta} | \psi \rangle}{\langle \psi | Q_{\beta} | \psi \rangle} \] (25)
\[ p_{\beta\alpha}^{b\mid a} = \frac{\langle \psi \mid P_a Q_{\beta} P_{\alpha} \mid \psi \rangle}{\langle \psi \mid P_{\alpha} \mid \psi \rangle} \quad (26) \]

hence, generally in the case of degenerate spectra, \( p_{\alpha\beta}^{a\mid b} \neq p_{\beta\alpha}^{b\mid a} \). It contradicts (10).

Now let us consider the case of nondegenerate spectra. In this case we can write two orthogonal projection operators as \( P_{\alpha} = |e_{\alpha}\rangle\langle e_{\alpha}| \) and \( Q_{\beta} = |e_{\beta}\rangle\langle e_{\beta}| \). Hence,

\[ p_{\alpha\beta}^{a\mid b} = \frac{\langle \psi \mid Q_{\beta} P_{a} Q_{\beta} \mid \psi \rangle}{\langle \psi \mid Q_{\beta} \mid \psi \rangle} = \frac{\langle \psi |e_{\beta}\rangle \langle e_{\beta}|e_{\alpha}\rangle \langle e_{\alpha}|e_{\beta}\rangle \langle e_{\beta}|\psi \rangle}{\langle \psi \mid Q_{\beta} \mid \psi \rangle} = \frac{\langle \psi |e_{\beta}\rangle \langle e_{\beta}|\psi \rangle (\langle e_{\alpha}|e_{\beta}\rangle)^2}{\langle \psi \mid Q_{\beta} \mid \psi \rangle} = |\langle e_{\alpha}|e_{\beta}\rangle|^2 \quad (27) \]

and

\[ p_{\beta\alpha}^{b\mid a} = \frac{\langle \psi \mid P_{a} Q_{\beta} P_{\alpha} \mid \psi \rangle}{\langle \psi \mid P_{\alpha} \mid \psi \rangle} = \frac{\langle \psi |e_{\alpha}\rangle \langle e_{\alpha}|e_{\beta}\rangle \langle e_{\beta}|e_{\alpha}\rangle \langle e_{\alpha}|\psi \rangle}{\langle \psi \mid P_{\alpha} \mid \psi \rangle} = \frac{\langle \psi |e_{\beta}\rangle \langle e_{\beta}|\psi \rangle (\langle e_{\beta}|e_{\alpha}\rangle)^2}{\langle \psi \mid P_{\alpha} \mid \psi \rangle} = |\langle e_{\beta}|e_{\alpha}\rangle|^2 \quad (28) \]

where \( |\langle e_{\alpha}|e_{\beta}\rangle|^2 = |\langle e_{\beta}|e_{\alpha}\rangle|^2 \). Finally, we obtain \( p_{\alpha\beta}^{a\mid b} = p_{\beta\alpha}^{b\mid a} \) in this case.

We remark that this inter-relation of symmetry implies that both matrices of transition probabilities are doubly stochastic.

Acknowledgements I am grateful to my supervisor Professor Andrei Khrennikov for many discussions on the formulations of quantum mechanics and for introducing me into this field of research. I am also very thankful to Guillaume Adenier for all the discussions we have had about the subject of quantum mechanics.

References

Paper II

On the consistency of the quantum-like representation algorithm for hyperbolic interference.

Peter Nyman

Published in: Advances in Applied Clifford Algebras, pages 1–13, 2010
On the Consistency of the Quantum-Like Representation Algorithm for Hyperbolic Interference

Peter Nyman

Abstract. Recently quantum-like representation algorithm (QLRA) was introduced by A. Khrennikov [20]–[28] to solve the so-called “inverse Born’s rule problem”: to construct a representation of probabilistic data by a complex or hyperbolic probability amplitude or more general complex together with hyperbolic which matches Born’s rule or its generalizations. The outcome from QLRA is coupled to the formula of total probability with an additional term corresponding to trigonometric, hyperbolic or hyper-trigonometric interference. The consistency of QLRA for probabilistic data corresponding to trigonometric interference was recently proved [29]. We complete the proof of the consistency of QLRA to cover hyperbolic interference as well. We will also discuss hyper trigonometric interference. The problem of consistency of QLRA arises, because formally the output of QLRA depends on the order of conditioning. For two observables (e.g., physical or biological) $a$ and $b$, $b|a$- and $a|b$-conditional probabilities produce two representations, say in Hilbert spaces $H_{b|a}$ and $H_{a|b}$ (in this paper over the hyperbolic algebra). We prove that under “natural assumptions” these two representations are unitary equivalent (in the sense of hyperbolic Hilbert space).

Keywords. Born’s rule problem, hyperbolic interference, hyper trigonometric interference, inverse order of conditioning, quantum-like representation algorithm.

1. Introduction

The interrelation between classical and quantum probabilities was early studied by von Neumann, see [1] and was followed by methods to generalize the probability theory to include quantum probabilities by Gudder, see [2]–[4]. For more recent and wide-ranging studies, see Svozil [5], [6], Fine [7], Garola et al. [8]–[10], Dvurecenskij and Pulmanova [12], Ballentine [11], O. Nánásiová et al. [13], [14], Allahverdyan et al. [15] and Khrennikov [30, 31]. The basic
The rule of quantum mechanics (QM) is the Born’s rule. Therefore the study of its origin is very important for quantum foundations. In a series of papers [20]–[28] Khrennikov studied the so called “inverse Born’s rule problem”:

**IBP** (inverse Born problem): To construct a representation of probabilistic data (of any origin) by a complex probability amplitude which matches Born’s rule.

The solution of IBP provides a possibility to represent probabilistic data by “wave functions” and apply this data by using linear algebra (as we do in conventional QM). However, as it was found in [20]–[28], some data do not permit the complex wave representation. In this case probabilistic amplitudes valued in the hyperbolic algebra (a two dimensional Clifford algebra) should be used. A special algorithm, quantum-like representation algorithm (QLRA), was created to transfer probabilities into probabilistic amplitudes. Depending on the data these amplitudes are complex, hyperbolic or hypercomplex.

Formally, the output of QLRA depends on the order of conditioning of probabilities. For two observables \(a\) and \(b\), \(b|a\) and \(a|b\) conditional probabilities produce two representations, for Hilbert spaces (complex or hyperbolic) \(H^{b|a}\) and \(H^{a|b}\). In this paper we will be interested in hyperbolic amplitudes as outputs of QLRA and therefore consider the hyperbolic Hilbert space. The case of complex amplitudes has already been studied in [20]. It was shown that under “natural assumptions” (non-degenerated assumptions) these two conditional probabilistic representations are unitary equivalent. This result proved the consistency of QLRA for complex amplitudes, now we will study the case of hyperbolic amplitudes.

In a purely mathematical framework the problem of consistency of two representations is nothing else than construction of a special unitary operator in hyperbolic Hilbert space establishing the equivalence of two representations. This paper is also a contribution to mathematical physics over hyperbolic numbers, see, e.g., [32]–[44].

### 2. Inversion of Born’s Rule

We consider the simplest situation. There are given two dichotomous observables of any context: \(a = \alpha_1, \alpha_2\) and \(b = \beta_1, \beta_2\). We set \(X_a = \{\alpha_1, \alpha_2\}\) and \(X_b = \{\beta_1, \beta_2\}\) — “spectra of observables”.

We assume that the matrix of transition probabilities is given by \(P^{b|a} = \langle p_{\beta \alpha}^{b|a} \rangle\), where \(p_{\beta \alpha}^{b|a} \equiv P(b = \beta|a = \alpha)\) is the probability to obtain the result \(b = \beta\) under the condition that the result \(a = \alpha\) has been obtained. There are also given probabilities \(p_{\alpha}^a \equiv P(a = \alpha), \alpha \in X_a\), and \(p_{\beta}^b \equiv P(b = \beta), \beta \in X_b\). Probabilistic data \(C = \{p_{\alpha}^a, p_{\beta}^b\}\) are related to some experimental context (in the physics preparation procedure).
On the Consistency of the Quantum-Like Representation Algorithm

IBP is to represent this data by a probability amplitude \( \psi \) (in the simplest case it is complex-valued, but we are interested in more general amplitudes) such that Born’s rule holds for both observables:

\[
p^b_\beta = |\langle \psi, e^b_\beta | \rangle|^2, \quad p^a_\alpha = |\langle \psi, e^a_\alpha | \rangle|^2,
\]

(2.1)

where \( \{e^b_\beta \}_{\beta \in X_b} \) and \( \{e^a_\alpha \}_{\alpha \in X_a} \) are orthonormal bases for observables \( b \) and \( a \), respectively (so the observables are diagonal in the respective bases).

In [20]–[28] the solution of IBP was given in the form of an algorithm which constructs a probability amplitude from the data. Formally, the output of this algorithm depends on the order of conditioning. By starting with the matrix of transition probabilities \( P^b|a \), instead of \( P^a|b \), we construct another probability amplitude \( \psi^a|b \) (the amplitude in (2.1) should be denoted by \( \psi^b|a \)) and other bases, \( \{e^a_\alpha \}_{\alpha \in X_a} \) and \( \{e^b_\beta \}_{\beta \in X_b} \). We shall see that under natural assumptions these two representations are unitary equivalent.

3. QLRA

3.1. \( H^b|a \)-conditioning

Suppose that the matrix of transition probabilities \( P^b|a \) is given. In [20]–[28] the following formula for the interference of probabilities (generalizing the classical formula of total probability) was derived:

\[
p^b_\beta = \sum_\alpha p^a_\alpha p^b|a_\beta \alpha + 2\lambda_\beta \sqrt{\prod_\alpha p^a_\alpha p^b|a_\beta \alpha},
\]

(3.1)

where the “coefficient of interference” is given by

\[
\lambda_\beta = \frac{p^b_\beta - \sum_\alpha p^a_\alpha p^b|a_\beta \alpha}{2\sqrt{\prod_\alpha p^a_\alpha p^b|a_\beta \alpha}}.
\]

(3.2)

We will proceed under the conditions:

1. \( P^b|a \) is doubly stochastic (for not doubly stochastic matrix, see section 5.3).
2. Probabilistic data \( C = \{p^a_\alpha, p^b_\beta \} \) consist of strictly positive probabilities.
3. The absolute values of the coefficients of interference \( \lambda_\beta, \beta \in X_b \), are larger than one: \( |\lambda_\beta| > 1 \).

Probabilistic data \( C \) such that \( |\lambda_\beta| \leq 1 \) are called trigonometric. In this case we have the conventional formula of trigonometric interference:

\[
p^b_\beta = \sum_\alpha p^a_\alpha p^b|a_\beta \alpha + 2\lambda_\beta \sqrt{\prod_\alpha p^a_\alpha p^b|a_\beta \alpha},
\]

where

\[
\lambda_\beta = \cos \theta_\beta.
\]

(3.3)
The case of trigonometric interference (i.e. \(|\lambda_\beta| \leq 1\)) has been studied in [29]. Therefore, now we consider the case of hyperbolic interference: \(|\lambda_\beta| > 1\). We represent this coefficient of interference by

\[
\lambda_\beta = \epsilon_\beta \cosh \theta_\beta ,
\]

(3.4)

where \(\epsilon_\beta = \text{sign } \lambda_\beta\).

Furthermore, in the case of hyper-trigonometric interference (i.e. \(|\lambda_\beta_i| > 1\) and \(|\lambda_\beta_j| \leq 1\)) we have \(\lambda_\beta_i = \cos \theta_\beta_i\) and \(\lambda_\beta_j = \epsilon_\beta_j \cosh \theta_\beta_j\), where \(\epsilon_\beta_j = \text{sign } \lambda_\beta_j\), \(i, j \in \{1, 2\}\), \(i \neq j\).

**Proposition 3.1.** Let \(P_{b|a}^{b|a}\) be doubly stochastic. Then the case of mixed hyper-trigonometric interference is excluded.

This proposition follows straightforward from the equality

\[
\lambda_\beta_1 + \lambda_\beta_2 = 0
\]

and the condition that \(P_{b|a}^{b|a}\) is doubly stochastic, see (3.2). We have:

\[
\lambda_\beta_1 + \lambda_\beta_2 = \frac{p_{b_1}^b - \sum_\alpha p_{a_\alpha} p_{b_{\beta_1}\alpha}^{b|a} + p_{b_2}^b - \sum_\alpha p_{a_\alpha} p_{b_{\beta_2}\alpha}^{b|a}}{2\sqrt{\prod_\alpha p_{a_\alpha} p_{b_{\beta_1}\alpha}^{b|a}}} + \frac{2\sqrt{\prod_\alpha p_{a_\alpha} p_{b_{\beta_2}\alpha}^{b|a}}}{2\sqrt{\prod_\alpha p_{a_\alpha} p_{b_{\beta_1}\alpha}^{b|a}}}
\]

\[
\begin{align*}
\lambda_\beta_1 + \lambda_\beta_2 &= \frac{p_{b_1}^b + p_{b_2}^b - \sum_\alpha p_{a_\alpha} p_{b_{\beta_1}\alpha}^{b|a} - \sum_\alpha p_{a_\alpha} p_{b_{\beta_2}\alpha}^{b|a}}{2\sqrt{\prod_\alpha p_{a_\alpha} p_{b_{\beta_1}\alpha}^{b|a}}} \\
&= 0 .
\end{align*}
\]

There is a contradiction between (3.5) and the definition of hyper-trigonometric interference: \(|\lambda_\beta_i| > 1\) and \(|\lambda_\beta_j| \leq 1\).

Therefore, we will focus on hyperbolic interference (since trigonometric interference has already been studied) and introduce the *hyperbolic algebra* \(G\); see appendix and [41]. Denote its generator (different from unit 1) by \(j\):

\[
j^2 = 1 .
\]

An element of \(G\) can be represented as \(z = x + jy\), \(x, y \in \mathbb{R}\). We introduce the hyperbolic exponential function

\[
e^{j\theta} = \cosh \theta + j \sinh \theta, \quad \theta \in \mathbb{R} .
\]

(3.6)

Define also \(\bar{z} = x - jy\), it is apparent that \(\bar{z} \in G\). We also use the identities

\[
cosh \theta = \frac{e^{j\theta} + e^{-j\theta}}{2}, \quad \sinh \theta = \frac{e^{j\theta} - e^{-j\theta}}{2j} .
\]

(3.7)

Thus, by using the elementary formula:

\[
D = A + B \pm 2\sqrt{AB} \cosh \theta = |\sqrt{A} \pm e^{j\theta} \sqrt{B}|^2, \quad A, B > 0, \ \theta \in \mathbb{R}, \ j^2 = 1
\]

(3.8)
for real numbers of $A$ and $B$, we can represent the probability $p^b_\beta$ as the square of the hyperbolic amplitude (Born’s rule): $p^b_\beta = |\psi^b_\beta|^2$. Here

$$\psi^b_\beta = \sqrt{p^b_{\alpha_1}p^b_{\alpha_1}} \pm e^{j\theta_\beta} \sqrt{p^b_{\alpha_2}p^b_{\alpha_2}}, \quad \beta \in X_b. \quad (3.9)$$

The formula (3.9) gives the hyperbolic amplitude, the output of QLRA for any probabilistic data $C$ if $|\lambda| > 1$. This is the normalized vector in the two dimensional hyperbolic Hilbert space\(^1\), say $H^{b|a}$:

$$\psi^{b|a} = \psi^{b|a} e^{b|a}_1 + \psi^{b|a} e^{b|a}_2, \quad (3.10)$$

where $e^{b|a}_1 = (1 \ 0)^T$, $e^{b|a}_2 = (0 \ 1)^T$.

To solve IBP completely, we would like to have Born’s rule not only for the $b$-variable, but also for the $a$-variable: $p^a_\alpha = |\langle \psi^{b|a}, e^{b|a}_\alpha \rangle|^2$, $\alpha \in X_a$. Here the $a$-basis in the hyperbolic Hilbert space $H^{b|a}$ is given, see [20]–[28] for details, by

$$e^{b|a}_{\alpha_1} = \left( \begin{array}{c} p^{b|a}_{\alpha_1} \\ \sqrt{p^{b|a}_{\alpha_1}} \end{array} \right), \quad e^{b|a}_{\alpha_2} = \left( \begin{array}{c} -p^{b|a}_{\alpha_2} \\ \sqrt{p^{b|a}_{\alpha_2}} \end{array} \right). \quad (3.11)$$

These basis vectors are orthonormal, since $P^{b|a}$ is assumed to be doubly stochastic. In this basis the hyperbolic amplitude $\psi^{b|a}$ is represented as

$$\psi^{b|a} = \sqrt{p^a_{\alpha_1}p^a_{\alpha_1}} \pm e^{j\theta_a} \sqrt{p^a_{\alpha_2}p^a_{\alpha_2}} e^{b|a}. \quad (3.12)$$

We recall that in QM two vectors (say $\psi'_1, \psi'_2$) define the same state $\psi'$ if they differ by multipliers of the form $c = e^{j\phi}$ (i.e. if $\psi'_1 = e^{j\phi} \psi'_2$ for some $\phi$). We will use a similar terminology for the case of the hyperbolic algebra: two vectors $\psi_1, \psi_2$ define the same state if $\psi_1 = \pm e^{j\gamma} \psi_2$. The consistency of this definition follows from the fact that

$$|\psi_2|^2 = |\pm e^{j\gamma}|^2 |\psi_2|^2 = |e^{j\gamma} \psi_2|^2 = |\psi_1|^2.$$ 

Thus measurements on these two states produce the same probability distribution.

Each hyperbolic amplitude $\psi^{b|a}$ produced by QLRA determines a quantum-like state (representing given probabilistic data) – the equivalence class $\Psi^{b|a}$ being determined by the representative $\psi^{b|a}$.

### 3.2. $H^{a|b}$-conditioning

For $a|b$-conditioning the state is represented by

$$\psi^{a|b}_\alpha = \sqrt{p^b_{\beta_1}p^a_{\alpha_1}} \pm e^{j\theta_a} \sqrt{p^b_{\beta_2}p^a_{\alpha_2}}, \quad \alpha \in X_a. \quad (13.3)$$

For any collection of probabilistic data $C$, QLRA produces the hyperbolic amplitude $\psi^{a|b}$ if $|\lambda| > 1$ (the normalized vector in the two dimensional hyperbolic Hilbert space, say $H^{a|b}$):

$$\psi^{a|b} = \psi^{a|b}_{\alpha_1} e^{a|b}_{\alpha_1} + \psi^{a|b}_{\alpha_2} e^{a|b}_{\alpha_2}, \quad (3.14)$$

\(^1\)For the definition of hyperbolic Hilbert space, see appendix.
where $e^{a|b}_{\beta_1} = (1 \ 0)^T$, $e^{a|b}_{\beta_2} = (0 \ 1)^T$. Here the $b$-basis in the hyperbolic Hilbert space $H^{a|b}$ is given by

$$e^{a|b}_{\beta_1} = \left(\sqrt{P^{a|b}_{\alpha_1 \beta_1}}, \sqrt{P^{a|b}_{\alpha_2 \beta_1}}\right), e^{a|b}_{\beta_2} = \left(-\sqrt{P^{a|b}_{\alpha_1 \beta_2}}, \sqrt{P^{a|b}_{\alpha_2 \beta_2}}\right). \quad (3.15)$$

In this basis the amplitude $\psi^{a|b}$ is represented as

$$\psi^{a|b} = \sqrt{p^{b}_{\beta_1}} e^{a|b}_{\beta_1} \pm e^{j\theta_{\alpha_1}} \sqrt{p^{b}_{\beta_2}} e^{a|b}_{\beta_2}. \quad (3.16)$$

As in the case of $H^{b|a}$-representation, the quantum-like state (representing given probabilistic data) is defined as the equivalence class $\Psi^{a|b}$ with the representative $\psi^{a|b}$.

4. Unitary Equivalence of $b|a$- and $a|b$-Representations

Thus, as we have seen, by selecting two types of conditioning, we represented the probabilistic data $C = \{p^a_\alpha, p^b_\beta\}$ by two quantum-like states, $\Psi^{b|a}$ and $\Psi^{a|b}$. We are interested in the consistency of these representations.

We remark that any linear operator $W : H^{b|a} \to H^{a|b}$ induces the map of equivalence classes of the hyperbolic unit sphere\footnote{The hyperbolic unit sphere is given by $|e^{j\theta}|^2 = \cosh^2 \theta - \sinh^2 \theta = 1$} with respect to multipliers $c = \pm e^{j\gamma}$. We define the unitary operator $U^{a|b}_{b|a} : H^{b|a} \to H^{a|b}$ by $U(e^{b|a}_\alpha) = e^{a|b}_\alpha, \alpha \in X_a$. It induces the mentioned map of equivalent classes.

**Theorem 4.1.** The operator $U^{a|b}_{b|a}$ maps $\Psi^{b|a}$ into $\Psi^{a|b}$ if and only if the following interrelation of symmetry takes place for the matrices of transition probabilities $P^{b|a}$ and $P^{a|b}$:

$$p^{b|a}_{\beta \alpha} = p^{a|b}_{\alpha \beta}, \quad (4.1)$$

for all $\alpha$ and $\beta$ from the spectra of observables $a$ and $b$. 

**Proof.** Take the representative of $\Psi^{b|a}$ given by (3.12). Then

$$U^{a|b}_{b|a} \psi^{b|a} = \sqrt{p^{a|b}_{\alpha_1 \beta_1}} e^{a|b}_{\alpha_1} \pm e^{j\theta_{\alpha_1}} \sqrt{p^{a|b}_{\alpha_2 \beta_2}} e^{a|b}_{\alpha_2}. \quad (4.2)$$

Our aim is to show that this vector is equivalent to the vector $\psi^{a|b}$ given by (3.14).

The coefficients of interference $\lambda_\alpha$ play in the $H^{a|b}$-representation the same role as the coefficients of interference $\lambda_\beta$ played in $H^{b|a}$-representation:

$$p^a_{\alpha_1} = p^b_{\beta_1} p^{a|b}_{\alpha_1 \beta_1} + p^b_{\beta_2} p^{a|b}_{\alpha_1 \beta_2} + \epsilon_{\alpha_1} 2|\lambda_{\alpha_1}| \sqrt{p^b_{\beta_1} p^{a|b}_{\alpha_1 \beta_1} p^b_{\beta_2} p^{a|b}_{\alpha_1 \beta_2}} \quad (4.3)$$

$$\Leftrightarrow$$

$$\lambda_{\alpha_1} = \frac{p^a_{\alpha_1} - p^b_{\beta_1} p^{a|b}_{\alpha_1 \beta_1} - p^b_{\beta_2} p^{a|b}_{\alpha_1 \beta_2}}{2\sqrt{p^b_{\beta_1} p^{a|b}_{\alpha_1 \beta_1} p^b_{\beta_2} p^{a|b}_{\alpha_1 \beta_2}}},$$

\footnote{The hyperbolic unit sphere is given by $|e^{j\theta}|^2 = \cosh^2 \theta - \sinh^2 \theta = 1$}
On the Consistency of the Quantum-Like Representation Algorithm

where \( \varepsilon_{\lambda_1} = \text{sign} \lambda_1 \). We consider \( |\lambda_1| > 1 \) and thus \( |\lambda_1| = \cosh \theta_1 \). We also calculate

\[
\psi_{\alpha_2}^{a|b} \psi_{\alpha_1}^{a|b} = p_{\beta_1}^b \sqrt{P_{\alpha_1 \beta_1}^a P_{\alpha_2 \beta_2}^b} + \varepsilon_{\lambda_1} \varepsilon_{\lambda_2} p_{\beta_2}^b \sqrt{P_{\alpha_2 \beta_2}^a P_{\alpha_1 \beta_1}^b} + \varepsilon_{\lambda_1} (\cosh \theta_1 + j \sinh \theta_1) \sqrt{p_{\beta_2}^b P_{\alpha_2 \beta_2}^a P_{\alpha_1 \beta_1}^b} + \varepsilon_{\lambda_1} (\cosh \theta_1 - j \sinh \theta_1) \sqrt{p_{\beta_2}^b P_{\alpha_1 \beta_1}^a P_{\alpha_2 \beta_2}^b},
\]

where \( \psi_{\alpha_2}^{a|b} = \sqrt{p_{\beta_1}^b P_{\alpha_2 \beta_1}^a} \mp e^{i \theta_1} \sqrt{p_{\beta_2}^b P_{\alpha_2 \beta_2}^a} \) is given by (3.16). We also use

\[
|\psi_{\alpha_i}^{a|b}|^2 = p_{\alpha_i}^a \Leftrightarrow \psi_{\alpha_i}^{a|b} = \pm \sqrt{p_{\alpha_i}^a} (\cosh \gamma_{\alpha_i} + j \sinh \gamma_{\alpha_i}),
\]

where\(^3\)

\[
\gamma_{\alpha_i} = \text{arg} \psi_{\alpha_i}^a, \ i \in \{1, 2\}
\]

This implies

\[
\psi_{\alpha_2}^{a|b} \psi_{\alpha_1}^{a|b} = \pm \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} (\cosh (\gamma_{\alpha_2} - \gamma_{\alpha_1}) + j \sinh (\gamma_{\alpha_2} - \gamma_{\alpha_1})). \tag{4.5}
\]

The real parts of the equations (4.4) and (4.5) give:

\[
\pm \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh (\gamma_{\alpha_2} - \gamma_{\alpha_1}) = p_{\beta_1}^b \sqrt{p_{\alpha_2 \beta_1}^a} - p_{\beta_2}^b \sqrt{p_{\alpha_2 \beta_2}^a} \tag{4.6}
\]

Notice that \( \lambda_{\alpha_2} = -\lambda_{\alpha_1} \) in (3.5) implies that

\[
\varepsilon_{\lambda_1} = -\varepsilon_{\lambda_2}, \varepsilon_{\lambda_1} \varepsilon_{\lambda_2} = -1.
\]

Moreover, since \( p_{\beta_2}^b = 1 - p_{\beta_1}^b \) and \( P_{\alpha|\beta}^a \) is doubly stochastic, i.e., \( p_{\alpha_2 \beta_1}^a = 1 - p_{\alpha_1 \beta_1}^a = 1 - p_{\alpha_2 \beta_2}^a \), we rewrite (4.6)

\[
\pm \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh (\gamma_{\alpha_2} - \gamma_{\alpha_1}) = (2p_{\beta_1}^b - 1) \sqrt{p_{\alpha_1 \beta_1}^a (1 - p_{\alpha_1 \beta_1}^a)} \tag{4.7}
\]

Then by (3.2) and (3.4) we obtain \( \cosh \theta_{\beta_1} \):

\[
\varepsilon_{\lambda_{\beta_1}} \cosh \theta_{\beta_1} = \frac{p_{\beta_1}^b - p_{\alpha_1 \beta_1}^a p_{\beta_1 \alpha_2}^b}{2 \sqrt{p_{\alpha_1 \beta_1}^a p_{\beta_1 \alpha_2}^b}}. \tag{4.8}
\]

Multiply (4.8) with \( 2 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \) and use again that \( p_{\alpha_2}^a = 1 - p_{\alpha_1}^a \) and \( P_{\alpha|\beta}^a \) is doubly stochastic and

\[
\varepsilon_{\lambda_{\beta_1}} 2 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh \theta_{\beta_1} = \frac{p_{\alpha_1}^a - 1 + p_{\beta_1}^b + p_{\alpha_1 \beta_1}^a - 2 p_{\alpha_1 \beta_1}^a p_{\alpha_1}^a}{\sqrt{p_{\alpha_1 \beta_1}^a p_{\beta_1 \alpha_2}^b}}. \tag{4.9}
\]

---

\(^3\)see appendix for definition of the argument (\( \text{arg} \)) in the hyperbolic algebra.
We will show that \( \pm \cosh (\gamma \alpha_2 - \gamma \alpha_1) = \epsilon \lambda \beta_1 \cosh \theta \beta_1 \) or equivalently, we show that
\[
\epsilon \lambda \beta_1 2 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh (\gamma \alpha_2 - \gamma \alpha_1) = 2 \epsilon \lambda \beta_1 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh \theta \beta_1. \tag{4.10}
\]
We multiply the left-hand side of (4.7) by \( 2 \sqrt{p_{\alpha_1}^a p_{\alpha_1}^b (1 - p_{\alpha_1}^a p_{\alpha_1}^b)} \). We get
\[
\text{LHS} = \pm 2 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh (\gamma \alpha_2 - \gamma \alpha_1) \text{ and replace } \\
\epsilon \lambda \alpha_1 \cosh \theta \alpha_1 \text{ by } \\
2 \sqrt{p_{\beta_1}^b p_{\alpha_1}^b p_{\alpha_2}^b} (1 - p_{\alpha_1}^a p_{\alpha_1}^b) \] in the right-hand side
\[
\text{LHS} = 2 \left( 2 p_{\beta_1}^b - 1 \right) p_{\alpha_1}^a (1 - p_{\alpha_1}^a) \\
+ \left( p_{\alpha_1}^a - p_{\beta_1}^b p_{\alpha_1}^b - (1 - p_{\beta_1}^b) (1 - p_{\alpha_1}^a) \right) \left( 1 - 2 p_{\alpha_1}^a \right) \tag{4.11}
\]
We calculate the last term:
\[
\left( p_{\alpha_1}^a - 1 + p_{\beta_1}^b + p_{\alpha_1}^b - 2 p_{\beta_1}^b p_{\alpha_1}^b \right) \left( 1 - 2 p_{\alpha_1}^a \right) \tag{4.12}
\]
or equivalently, we show
\[
\epsilon \lambda \beta_1 2 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh (\gamma \alpha_2 - \gamma \alpha_1) = 2 \epsilon \lambda \beta_1 \sqrt{p_{\alpha_1}^a p_{\alpha_2}^a} \cosh \theta \beta_1. \tag{4.10}
\]

Moreover,
\[
\text{LHS} = 2 \left( 2 p_{\beta_1}^b - 1 \right) p_{\alpha_1}^a (1 - p_{\alpha_1}^a) \\
+ \left( p_{\alpha_1}^a - 1 + p_{\beta_1}^b + p_{\alpha_1}^b - 2 p_{\beta_1}^b p_{\alpha_1}^b \right) \left( 1 - 2 p_{\alpha_1}^a \right) \tag{4.11}
\]

Equations (4.9) and (4.13) imply that
\[
\pm p_{\alpha_1}^a - 1 + p_{\beta_1}^b + p_{\alpha_1}^b - 2 p_{\alpha_1}^b p_{\alpha_1}^a \sqrt{p_{\alpha_1}^a p_{\beta_1}^a} = \pm p_{\alpha_1}^a - 1 + p_{\beta_1}^b + p_{\alpha_1}^b - 2 p_{\beta_1}^b p_{\alpha_1}^a \\
\sqrt{p_{\beta_1}^a p_{\alpha_1}^a} \tag{4.14}
\]

\[
implies \sqrt{p_{\alpha_1}^a p_{\beta_1}^a} = \sqrt{p_{\alpha_1}^a p_{\beta_1}^a}. 
\]
On the Consistency of the Quantum-Like Representation Algorithm

Therefore we conclude that \( \pm \cosh(\gamma_{\alpha_2} - \gamma_{\alpha_1}) = \epsilon \lambda \beta_1 \cosh(\gamma_{\alpha_1}) \) if and only if \( P^{b|a} = P^{a|b} \).

Let

\[
U^{a|b}_{b|a} = \left( \begin{array}{cc} \sqrt{P_{b\alpha_1}} & \sqrt{P_{b\alpha_2}} \\ \sqrt{P_{b\beta_1}} & -\sqrt{P_{b\beta_2}} \end{array} \right). \tag{4.15}
\]

We now show that this vector is equivalent to the vector \( \psi^{a|b} \) given by (3.14).

\[
U^{a|b}_{b|a} \psi^{b|a} = \sqrt{P_{a\alpha_1}} e^{a|b} + \epsilon \lambda \beta_1 \cosh(\gamma_{\alpha_1}) \sqrt{P_{a\beta_1}} e^{a|b} \tag{4.16}
\]

We use the fact that \( \psi^{a|b}_{\alpha_i} = \pm \sqrt{P_{a\alpha_i}} e^{j\alpha_i}, i \in \{1, 2\} \) into (3.14)

\[
\psi^{a|b} = \pm \sqrt{P_{a\alpha_1}} e^{j\alpha_1} e^{a|b} \pm \sqrt{P_{a\alpha_2}} e^{j\alpha_2} e^{a|b} \tag{4.17}
\]

Thus the hyperbolic amplitudes \( \psi^{a|b} \) and \( U^{a|b}_{b|a} \psi^{b|a} \) differ only by the multiplicative factor \( \pm e^{j\alpha_1} \). Hence, they belong to the same equivalent class of vectors on the unit sphere. Thus they are two representatives of the same quantum state \( \Psi^{b|a} \).

5. Appendix: Hyperbolic Algebra and Hyperbolic Hilbert Space

5.1. Hyperbolic algebra

An element \( z \) belongs to the hyperbolic algebra \( \mathbb{G} \) if and only if it has the following form:

\[
z = x + jy, \quad x, y \in \mathbb{R}
\]

where \( j^2 = 1 \), \( z_1 + z_2 = x_1 + x_2 + j(y_1 + y_2) \) and \( z_1 z_2 = x_1 x_2 + y_1 + y_2 + j(y_1 x_2 + y_2 x_1) \). The hyperbolic conjugation is defined as \( \bar{z} = x - jy \). We define the “square of the absolute value” as

\[
|z|^2 = z\bar{z} = x^2 - y^2,
\]

\( |z|^2 \in \mathbb{G} \). In fact, \( |z|^2 \in \mathbb{R} \). But \( |z| \) is not well defined for \( z \) such that \( |z|^2 < 0 \). Therefore set

\[
\mathbb{G}_+ = \{ z \in \mathbb{G} : |z|^2 \geq 0 \}.
\]

and

\[
\mathbb{G}^*_+ = \{ z \in \mathbb{G} : |z|^2 > 0 \}.
\]

We define the argument \( \arg z \) of \( z \in \mathbb{G}^*_+ \) as

\[
\arg z = \arctanh \frac{y}{x} = \frac{1}{2} \ln \frac{x + y}{x - y}.
\]

Notice that \( x \neq 0 \), \( x - y \neq 0 \) and \( \frac{x + y}{x - y} > 0 \), since \( z \in \mathbb{G}^*_+ \).
5.2. Hyperbolic Hilbert space

A hyperbolic Hilbert space $H$ is a $G$-linear inner product space. Let $x, y, z \in H$ and $a, b \in G$, then consider the inner product as a map from $H \times H$ to $G$ having the following properties:

1. Conjugate symmetry: $\langle x, y \rangle$ is the conjugate to $\langle y, x \rangle$

$$\langle x, y \rangle = \overline{\langle y, x \rangle}$$

2. Linearity with respect to the first argument:

$$\langle ax + bz, y \rangle = a \langle x, y \rangle + b \langle z, y \rangle$$

3. Nondegenerate:

$$\langle x, y \rangle = 0$$

for all $y \in H$ if and only if $x = 0$

In general, the norm $\| \psi \| = \sqrt{\langle \psi, \psi \rangle}$ is not well defined. But we will only need the square of the norm $\| \psi \|^2 = \langle \psi, \psi \rangle$.

5.3. Violation of Born’s rule

Let us give a counter example to illustrate the violation of Born’s rule, if the transition probabilities matrix $P^{[b|a]}$ is not doubly stochastic. We have that

$$\psi^{[b|a]}_\beta = \sqrt{P^a_{\alpha_1} P^{[b|a]}_{\beta \alpha_1}} \pm e^{i\theta_\beta} \sqrt{P^a_{\alpha_2} P^{[b|a]}_{\beta \alpha_2}}, \quad \beta \in X_b$$

and

$$\psi^{[b|a]} = \psi^{[b|a]}_{\beta_1} e^{b|a}_{\beta_1} + \psi^{[b|a]}_{\beta_2} e^{b|a}_{\beta_2}.$$ (5.2)

This will match Born’s rule,

$$p^b_\beta = |\langle \psi^{[b|a]}_\beta, e^{b|a}_\beta \rangle|^2, \quad \beta \in X_b.$$ (5.3)

Moreover $p^b_{\beta_1} + p^b_{\beta_2} = 1$ and by (3.1),

$$1 = p^b_{\beta_1} + p^b_{\beta_2} = p^a_{\alpha_1} (P^{[b|a]}_{\beta_1 \alpha_1} + P^{[b|a]}_{\beta_2 \alpha_1}) + p^a_{\alpha_2} (P^{[b|a]}_{\beta_1 \alpha_2} + P^{[b|a]}_{\beta_2 \alpha_2})$$

$$+ 2 \sqrt{P^a_{\alpha_1} P^a_{\alpha_2}} (\lambda_1 \sqrt{P^{[b|a]}_{\beta_1 \alpha_1} P^{[b|a]}_{\beta_1 \alpha_2}} + \lambda_2 \sqrt{P^{[b|a]}_{\beta_1 \alpha_2} P^{[b|a]}_{\beta_2 \alpha_2}}).$$ (5.4)

Let us select the transition probabilities matrix $P^{[b|a]}$ not to be doubly stochastic. Take $p^{[b|a]}_{\beta_1 \alpha_2} = p^{[b|a]}_{\beta_1 \alpha_1} = p$ and $p^{[b|a]}_{\beta_2 \alpha_1} = p^{[b|a]}_{\beta_2 \alpha_2} = q$ where $p + q = 1$, $p \neq q$, $p, q > 0$.

Then (5.4) becomes

$$1 = p^a_{\alpha_1} + p^a_{\alpha_2} + 2 \sqrt{P^a_{\alpha_1} P^a_{\alpha_2}} (\lambda_1 p + \lambda_2 q) \iff \lambda_1 = -\frac{q}{p} \lambda_2$$ (5.5)

Then (3.12) and (3.11) will be

$$\psi^{[b|a]} = \sqrt{p^a_{\alpha_1} e^{b|a}_{\alpha_1}} \pm e^{i\theta_\beta_1} \sqrt{p^a_{\alpha_2} e^{b|a}_{\alpha_2}}$$ (5.6)

where

$$e^{b|a}_{\alpha_1} = \left( \frac{\sqrt{p^{[b|a]}_{\beta_1 \alpha_1}}}{P^{[b|a]}_{\beta_1 \alpha_1}}, \frac{\sqrt{p^{[b|a]}_{\beta_1 \alpha_2}}}{P^{[b|a]}_{\beta_2 \alpha_1}} \right), \quad e^{b|a}_{\alpha_2} = \left( \frac{\sqrt{p^{[b|a]}_{\beta_2 \alpha_2}}}{P^{[b|a]}_{\beta_2 \alpha_2}} \right).$$ (5.7)
On the Consistency of the Quantum-Like Representation Algorithm

Thus \( p^b_\beta = |\langle e^b_{\alpha_1}, e^b_{\alpha_2} \rangle|^2 = p - \frac{q^2}{p} \), where \( e^b_{\alpha_1} \) and \( e^b_{\alpha_2} \), are orthogonal, i.e. \( |\langle e^b_{\alpha_1}, e^b_{\alpha_2} \rangle|^2 = 0 \). This shows the violation of Born’s rule by contradiction, since \( p - \frac{q^2}{p} = 0 \iff p = \pm q \).

Acknowledgment

I am grateful to my supervisor Professor Andrei Khrennikov for discussions and introducing me into this field of research. I am also very thankful to Guillaume Adenier for discussions on foundations of quantum mechanics.

References


On the Consistency of the Quantum-Like Representation Algorithm


Peter Nyman
School of Computer Science,
Physics and Mathematics
Linnaeus University, S-35195
Sweden
e-mail: peter.nyman@lnu.se

Received: January 14, 2010.
Revised: October 19, 2010.
Accepted: January 13, 2011.
Paper III

Representation of probabilistic data by complex probability amplitudes; the case of triple-valued observables.

Peter Nyman and Irina Basieva.

Representation of probabilistic data by complex probability amplitudes; the case of triple–valued observables

Peter Nyman and Irina Basieva

International Center for Mathematical Modelling in Physics and Cognitive Sciences, Linnaeus University, S-35195, Sweden

Abstract. We study the problem of representation of statistical data (of any origin) by a complex probability amplitude. This paper is devoted to representation of data collected from measurements of two trichotomous observables.

Keywords: Born’s rule, Probabilistic data, Quantum-like, Trichotomous observables

PACS: 02.50.Cw, 02.50.Tt, 03.65.Ca, 03.65.Ta

INTRODUCTION

The problem of inter-relation between classical and quantum probabilistic data was discussed in numerous papers (from various points of view), see, e.g., [1, 2, 3, 4, 6, 5, 7, 8, 14, 15]. We are interested in the problem of representation of probabilistic data of any origin ¹ by complex probability amplitude, so to say a “wave function”. This problem was discussed in very detail in [17]. A general QL-representation algorithm (QLRA) was presented in [17]. This algorithm is based on the formula of total probability with interference term – a disturbance of the standard formula of total probability. Starting with experimental probabilistic data, QLRA produces a complex probability amplitude such that probability can be reconstructed by using Born’s rule.

Although the formal scheme of QLRA works for multi-valued observables of an arbitrary dimension, the description of the class of probabilistic data which can be transferred into QL-amplitudes (the domain of application of QLRA) depends very much on the dimension. In [19] the simplest case of data generated by dichotomous observables was studied. In this paper we study trichotomous observables. The complexity of the problem increases incredibly comparing with the two dimensional case.

Finally, we remark that our study is closely related to the triple slit interference experiment and Sorkin’s equality [16]. This experiment provides an important test of foundations of QM.

The scheme of presentation is the following one. We start with observables given by QM and derive constraints on phases which are necessary and sufficient for the QL-representation. Then we use these constraints to produce complex amplitudes from data (of any origin); some examples, including numerical, are given.

¹ Thus it need not be produced by quantum measurements; it can be collected in e.g. psychology, see [18].
TRICHOTOMOUS INCOMPATIBLE QUANTUM OBSERVABLES

Probabilities

Let $a$ and $b$ be two self-adjoint operators in three dimensional complex Hilbert space representing two trichotomous incompatible observables $a$ and $b$. They take values $a = \alpha_i$, $i = 1, 2, 3$ and $b = \beta_i$, $l = 1, 2, 3$ – spectra of operators. We assume that the operators have nondegenerate spectra, i.e., $\alpha_i \neq \alpha_j, \beta_i \neq \beta_j, i \neq j$. Consider corresponding eigenvectors:

$$\hat{a} e^{a}_{\alpha_i} = \alpha_i e^{a}_{\alpha_i}, \hat{b} e^{b}_{\beta_i} = \beta_i e^{b}_{\beta_i}.$$

Denote by $\hat{P}^{a}_{\alpha_i} = | e^{a}_{\alpha_i} \rangle \langle e^{a}_{\alpha_i} |$ and $\hat{P}^{b}_{\beta_i} = | e^{b}_{\beta_i} \rangle \langle e^{b}_{\beta_i} |$ one dimensional projection operators and by $P^{a}_{\alpha_i}$ and $P^{b}_{\beta_i}$ the observables repressed by there projections. Consider also projections

$$P^{a\perp}_{\alpha_i} = P^{a}_{\alpha_i} + P^{a\perp}_{\alpha_i},$$

$$P^{a}_{\alpha_i} = P^{a}_{\alpha_1} + P^{a}_{\alpha_2},$$

$$P^{a}_{\alpha_3} = P^{a}_{\alpha_1} + P^{a}_{\alpha_2}.$$

Here the observable $P^{a}_{\alpha_i}$ is if the result of the $a$-measurement is $a = \alpha_i$, and $P^{a}_{\alpha_i} = 0$ if $a \neq \alpha_i$. The observables $P^{b}_{\beta_i}$ are defined in the same way. We have the following relation between events corresponding to measurements $[P^{a}_{\alpha_i} = 0] = [P^{a}_{\alpha_2} = 1] \lor [P^{a}_{\alpha_3} = 1]$, $[P^{a}_{\alpha_i} = 0] = [P^{a}_{\alpha_1} = 1] \lor [P^{a}_{\alpha_2} = 1]$. There are given (by the QM-formalism) the probabilities

$$p^{b}_{\beta} \equiv P_{\psi}(b = \beta) = ||\hat{P}^{b}_{\beta} \psi||^2 = ||\langle \psi | e^{b}_{\beta} \rangle ||^2,$$

$$p^{a}_{\alpha} \equiv P_{\psi}(a = \alpha) = ||\hat{P}^{a}_{\alpha} \psi||^2 = ||\langle \psi | e^{a}_{\alpha} \rangle ||^2.$$

There are also given (by the QM-formalism) conditional (transition) probabilities

$$p^{b|a}_{\beta \alpha} \equiv P_{\psi}(b = \beta | P^{a}_{\alpha_i} = 1) = ||\hat{P}^{b}_{\beta} \hat{P}^{a}_{\alpha} \psi||^2 / ||\hat{P}^{a}_{\alpha} \psi||^2 = ||\langle e^{b}_{\beta} | e^{a}_{\alpha} \rangle ||^2.$$

We remark that non degeneration of the spectra implies that they do not depend on $\psi$. Moreover, the matrix of transition probabilities is doubly stochastic. There are also given ($\psi$-depend) probabilities

$$p^{b|a}_{\beta \alpha_{ij}} \equiv P_{\psi}(b = \beta_i | P^{a}_{\alpha_{j}} = 1 \lor P^{a}_{\alpha_j} = 1) = P_{\psi}(b = \beta_i | P^{a}_{\alpha_j} = 0) = \frac{||\hat{P}^{b}_{\beta_i} \hat{P}^{a}_{\alpha_{i}} + \hat{P}^{a}_{\alpha_{j}} \psi||^2}{||\hat{P}^{a}_{\alpha_{j}} + \hat{P}^{a}_{\alpha_{i}} \psi||^2},$$

where $i, j, k \in \{1, 2, 3\}$, $i, j \neq k$. We have

$$\frac{||\hat{P}^{b}_{\beta_i} (\hat{P}^{a}_{\alpha_{j}} + \hat{P}^{a}_{\alpha_{i}}) \psi||^2}{||\hat{P}^{a}_{\alpha_{j}} + \hat{P}^{a}_{\alpha_{i}} \psi||^2} = \frac{||\langle \psi | e^{b}_{\beta_i} | e^{a}_{\alpha_{i}} \rangle | e^{a}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle + | e^{b}_{\beta_i} \rangle \langle e^{b}_{\alpha_{i}} | e^{b}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle ||^2}{||\langle \psi | e^{a}_{\alpha_{i}} \rangle | e^{a}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle + | e^{b}_{\beta_i} \rangle \langle e^{b}_{\alpha_{i}} | e^{b}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle ||^2}\frac{||\langle \psi | e^{a}_{\alpha_{i}} \rangle | e^{a}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle + | e^{b}_{\beta_i} \rangle \langle e^{b}_{\alpha_{i}} | e^{b}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle ||^2}{||\langle \psi | e^{a}_{\alpha_{i}} \rangle | e^{a}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle + | e^{b}_{\beta_i} \rangle \langle e^{b}_{\alpha_{i}} | e^{b}_{\alpha_{j}} \rangle \langle e^{a}_{\alpha_{j}} | \psi \rangle ||^2}.$$
Probability amplitudes

Set \( \psi_B = \langle \psi | e^b \rangle \). Then by Born’s rule

\[
p_B^b = |\psi_B|^2. \tag{4}
\]

We have

\[
\psi = \sum_B \psi_B e^b. \tag{5}
\]

Thus these amplitudes give a possibility to reconstruct the state. We remark that \( \psi = \sum_\alpha \hat{p}_\alpha \psi \), and, hence,

\[
\psi_B = \sum_\alpha \langle \hat{p}_\alpha^* \psi | e^b \rangle. \tag{6}
\]

Each amplitude \( \psi_B \) can be represented as the sum of three subamplitudes

\[
\psi_B = \sum_\alpha \psi_{B\alpha} \tag{7}
\]

given by

\[
\psi_{B\alpha} = \langle \hat{p}_\alpha^* \psi | e^b \rangle = \langle \psi | e^a \rangle \langle e^a | e^b \rangle. \tag{8}
\]

Hence, one can reconstruct the state \( \psi \) on the basis of nine amplitudes \( \psi_{B\alpha} \). We remark that

\[
|\psi_{B\alpha}|^2 = |\langle \psi | e^a \rangle \langle e^a | e^b \rangle|^2 = \hat{p}_\alpha^a \hat{p}_\alpha^b. \tag{9}
\]

In this notations

\[
P_{\alpha\alpha} = |\psi_{\alpha\alpha}|^2 / (\hat{p}_\alpha^a + \hat{p}_\alpha^b).
\]

Here \( |\psi_{\alpha\alpha}| = \sqrt{\hat{p}_\alpha^a \hat{p}_\alpha^b} \) and therefore

\[
\psi_{\alpha\alpha} = \sqrt{\hat{p}_\alpha^a \hat{p}_\alpha^b} e^{i\phi_{\alpha\alpha}},
\]

where \( \phi_{\alpha\alpha} = \arg \psi_{\alpha\alpha}. \) Moreover,

\[
\langle \psi | e^a \rangle = \sqrt{\hat{p}_\alpha^a} e^{i\phi_{\alpha\alpha}}, \quad \langle e^a | e^b \rangle = \sqrt{\hat{p}_\alpha^b} e^{i\phi_{\alpha\alpha}}. \tag{10}
\]

Hence,

\[
\phi_{\alpha\alpha} = \theta_{\alpha\alpha} + \xi_{\alpha\alpha}. \tag{11}
\]

We have a system of equations for phases \( \psi_{\alpha\alpha} \) for \( i, j, k, l \in \{1, 2, 3\} \),

\[
|\psi_{\alpha\alpha} + \psi_{\beta\alpha}|^2 = \sqrt{\hat{p}_\alpha^a \hat{p}_\alpha^b} e^{i\phi_{\alpha\alpha}} + \sqrt{\hat{p}_\alpha^b \hat{p}_\alpha^a} e^{i\phi_{\beta\alpha}} \tag{12}
\]

\[
= \hat{p}_\alpha^a \hat{p}_\alpha^b + \hat{p}_\alpha^b \hat{p}_\alpha^a
\]

\[
+ 2 \cos(\phi_{\alpha\alpha} - \phi_{\beta\alpha}) \sqrt{\hat{p}_\alpha^a \hat{p}_\alpha^b \hat{p}_\beta^a \hat{p}_\beta^b}.
\]
We set
\[ \lambda_{l,i j} \equiv \cos(\varphi_{\beta l \alpha_i} - \varphi_{\beta l \alpha_j}) \]  
(13)
and we have
\[ \lambda_{l,i j} = \frac{(p_{\alpha i}^{a} + p_{\alpha j}^{a}) P_{\beta l \alpha_i}^{b i a} - (p_{\alpha i}^{a} P_{\beta l \alpha_i}^{a} + p_{\alpha j}^{a} P_{\beta l \alpha_i}^{b i a})}{2\sqrt{p_{\alpha i}^{a} P_{\beta l \alpha_i}^{b i a} P_{\alpha l \alpha_j}^{b i a}}}. \]  
(14)
We call \( \lambda_{l,i j} \) for the coefficients of interference.

**Formula of total probability with interference term**

By using the definition of the amplitude \( \psi_{\beta l \alpha_i} = \langle \psi | e_{\alpha_i}^{a} \rangle \langle e_{\alpha_i}^{b} | e_{\beta}^{b} \rangle \) we obtain
\[
p_{\beta l}^{b} = |\psi_{\beta l \alpha_i} + \psi_{\beta l \alpha_j} + \psi_{\beta l \alpha_k}|^2 = |\langle \psi | e_{\alpha_i}^{a} \rangle \langle e_{\alpha_i}^{b} | e_{\beta}^{b} \rangle + \langle \psi | e_{\alpha_j}^{a} \rangle \langle e_{\alpha_j}^{b} | e_{\beta}^{b} \rangle + \langle \psi | e_{\alpha_k}^{a} \rangle \langle e_{\alpha_k}^{b} | e_{\beta}^{b} \rangle |^2.
\]
(15)
Finally, we obtain
\[
p_{\beta l}^{b} = p_{\alpha l}^{a} p_{\beta l \alpha_i}^{b i a} + p_{\alpha l}^{a} p_{\beta l \alpha_j}^{b i a} + p_{\alpha l}^{a} p_{\beta l \alpha_k}^{b i a} + 2\cos(\varphi_{\beta l \alpha_i} - \varphi_{\beta l \alpha_j}) \sqrt{p_{\alpha l}^{a} p_{\alpha l}^{b i a} + 2\cos(\varphi_{\beta l \alpha_i} - \varphi_{\beta l \alpha_j}) \sqrt{p_{\alpha l}^{a} P_{\beta l \alpha_i}^{b i a} + \sqrt{p_{\alpha l}^{a} P_{\beta l \alpha_j}^{b i a}}}}.
\]
(16)
This is nothing else than the formula of total probability with the interference term. It can be considered [21] as a perturbation of the classical formula of total probability
\[
p_{\beta l}^{b} = p_{\alpha l}^{a} p_{\beta l \alpha_i}^{b i a} + p_{\alpha l}^{a} p_{\beta l \alpha_j}^{b i a} + p_{\alpha l}^{a} p_{\beta l \alpha_k}^{b i a}.
\]
(17)
If all coefficients of interferes \( \lambda_{l,i j} = 0 \), then (16) coincides with (17).
Sorkin’s equality in conditional probabilistic form

We will derive Sorkin’s equality by putting (13) in (16),

\[ p^b_{\beta_l} = p^a_\alpha P^b_{\beta_l \alpha_i} + p^a_\alpha P^b_{\beta_l \alpha_j} + p^a_\alpha P^b_{\beta_l \alpha_k} + \left( (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_i} - (p^a_\alpha P^b_{\beta_l \alpha_i} + p^a_\alpha P^b_{\beta_l \alpha_i}) \right) \]

\[ + \left( (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_j} + p^a_\alpha P^b_{\beta_l \alpha_j} + p^a_\alpha P^b_{\beta_l \alpha_j} \right) \]

\[ + \left( (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_k} + p^a_\alpha P^b_{\beta_l \alpha_k} + p^a_\alpha P^b_{\beta_l \alpha_k} \right) \]

\[ = (p^a_\alpha + p^a_\alpha) P^b_{\beta_l \alpha_i} + p^a_\alpha P^b_{\beta_l \alpha_j} + (p^a_\alpha + p^a_\alpha) P^b_{\beta_l \alpha_k} + (p^a_\alpha + p^a_\alpha) P^b_{\beta_l \alpha_i} \]

\[ - (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_j} - (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_k} - (p^a_\alpha + p^a_\alpha) p^b_{\beta_l \alpha_i} \]

\[ + p^a_\alpha P^b_{\beta_l \alpha_j} + p^a_\alpha P^b_{\beta_l \alpha_k} - p^a_\alpha P^b_{\beta_l \alpha_i} \].

This gives us the following constraint on the probabilities

\[ p^b_{\beta_l} = p^a_\alpha (P^b_{\beta_l \alpha_i} + P^b_{\beta_l \alpha_j} - P^b_{\beta_l \alpha_i}) \]

\[ + p^a_\alpha (P^b_{\beta_l \alpha_j} + P^b_{\beta_l \alpha_k} - P^b_{\beta_l \alpha_i}) + p^a_\alpha (P^b_{\beta_l \alpha_k} + P^b_{\beta_l \alpha_i} - P^b_{\beta_l \alpha_i}) \].

This equation coupling various quantum probabilities can be considered as encrypting of Born’s rule by using the language of probabilities. This is the discrete version of famous Sorkin equality [22, 23].

CONSTRUCTION OF A COMPLEX PROBABILITY AMPLITUDE SATISFYING BORN’S RULE

Now we have a pair of trichotomous observables \( a \) and \( b \) taking values \( a = \alpha_i, \ i = 1, 2, 3 \) and \( b = \beta_l, l = 1, 2, 3 \). We do not assume that they have any relation to quantum physics; e.g., these are some random variables observed in biology or finances. It is assumed that there are given probability distributions of these variables

\[ p^b_{\beta_l} = P(b = \beta_l), \ p^a_\alpha = P(a = \alpha_i). \]

Thus

\[ \sum_{l=1}^{3} p^b_{\beta_l} = 1, \sum_{i=1}^{3} p^a_\alpha = 1. \]  \hspace{1cm} (20)

It is also assumed that there are given conditional probabilities \( p^{b|a}_{\beta_l \alpha_i} = P(b = \beta_l|a = \alpha_i) \).

We know that for any sort of data the matrix of transition probabilities is stochastic, i.e., for each \( \alpha_i \)

\[ \sum_{l=1}^{3} P^{b|a}_{\beta_l \alpha_i} = 1. \]  \hspace{1cm} (21)
Finally, we assume a possibility to collect the data on measurements of observables $P_{\alpha_i}^a, i = 1, 2, 3$, probabilities $p_{\beta_i \alpha_j}^{b/a} = P(b = \beta_i | P_{\alpha_i}^a = 0)$. “The detector corresponding to the value $a = \alpha_i$ does not click, so the value of $a$ is either $a = \alpha_j$ or $a = \alpha_k$, where $j, k \neq i$. However, we do not know the value of $a$. In this context we measure the $b$-variable.” For any sort of data, we have

$$\sum_{i=1}^{3} p_{\beta_i \alpha_j}^{b/a} = 1. \quad (22)$$

**Complex amplitude matching Born’s rule for one observable**

Now we want to find a complex probability amplitude $\psi_{\beta_i}$ such that Born’s rule (for the $b$-variable) holds: $|\psi_{\beta_i}|^2 = p_b^{b/a}$. We copy the QM-scheme, so we represent $\psi_{\beta_i} = \psi_{\beta_i \alpha_1} + \psi_{\beta_i \alpha_2} + \psi_{\beta_i \alpha_3}$, where the sub-amplitudes $\psi_{\beta_i \alpha_1} = \sqrt{P_{\alpha_i}^a p_{\beta_i \alpha_1}^{b/a}} e^{i \phi_{\beta_i \alpha_1}}$ and phases are determined by the system of equations (13). It is convenient to work with the interference coefficients, see [17], given by right-hand sides of these equations

$$\lambda_{i,j} = \frac{(p_{\alpha_i}^a + p_{\alpha_j}^a) p_{\beta_i \alpha_j}^{b/a} - (p_{\alpha_i}^a p_{\beta_i \alpha_1}^{b/a} + p_{\alpha_j}^a p_{\beta_i \alpha_j}^{b/a})}{2 \sqrt{p_{\alpha_i}^a p_{\beta_i \alpha_1}^{b/a} p_{\alpha_j}^a p_{\beta_i \alpha_j}^{b/a}}} \quad (23)$$

Interference coefficients obtained in quantum physics are always bounded by 1:

$$|\lambda_{i,j}| \leq 1. \quad (24)$$

However, since we start with data of any origin, the condition (24) has to be checked to proceed to representation of data by complex amplitudes.\(^2\) If the system of equations, $m = 1, 2, 3$,

$$\cos(\phi_{\beta_i \alpha_1} - \phi_{\beta_m \alpha_2}) = \lambda_{m,12}, \quad (25)$$
$$\cos(\phi_{\beta_i \alpha_2} - \phi_{\beta_m \alpha_3}) = \lambda_{m,23},$$
$$\cos(\phi_{\beta_i \alpha_1} - \phi_{\beta_m \alpha_3}) = \lambda_{m,13},$$

has a solution (three phases) then we can construct the probability amplitudes $\psi_{\beta_i \alpha}$ and, hence, the probability amplitudes $\psi_{\beta_i}$ and the corresponding vector $\psi$.

However, in general such amplitudes will not provide a solution of the “inverse Born problem”, namely, Born’s rule can be violated. To obtain the real solution one should solve the system (25) under the constraint (19). Thus, to proceed toward a proper complex amplitude, one should first check the validity of (19) and then to solve the system (25). It is convenient to express “triple probabilities” $p_{\beta_m \alpha_j}$ through coefficients of interference

$$p_{\beta_m \alpha_j} = \frac{1}{p_{\alpha_i}^a + p_{\alpha_j}^a} (p_{\alpha_i} p_{\beta_m \alpha_j} + p_{\alpha_j} p_{\beta_m \alpha_j} - 2 \lambda_{\beta_m \alpha_j} \sqrt{p_{\alpha_i} p_{\beta_m \alpha_j} p_{\alpha_j} p_{\beta_m \alpha_j}}). \quad (26)$$

\(^2\) If this condition is violated then data may be represented by so called hyperbolic probabilistic amplitudes [20].
We remark that if (24) holds, then triple probabilities given by (26) are always nonnegative. By using the $\lambda$-variables normalization equations (22) can be written as $(j, k = 1, 2, 3)$

$$\sum_{l=1}^{3} \lambda_{l,jk} \sqrt{p_{\beta_l\alpha_j} p_{\beta_k\alpha_k}} = 0. \quad (27)$$

We also can write Sorkin’s equality (in fact, the formula of total probability with interference terms) as

$$p^{b}_{\beta_l} = p^{a}_{\alpha_i} p^{b|a}_{\beta_l\alpha_i} + p^{a}_{\alpha_i} p^{b|a}_{\beta_j\alpha_j} + p^{a}_{\alpha_i} p^{b|a}_{\beta_k\alpha_k} + 2\lambda_{l,ij} \sqrt{p_{\alpha_j} p^{a}_{\alpha_j} p^{b|a}_{\beta_l\alpha_i} p^{b|a}_{\beta_l\alpha_j}}$$

$$+ 2\lambda_{l,jk} \sqrt{p^{a}_{\alpha_j} p^{a}_{\alpha_k} p^{b|a}_{\beta_l\alpha_j} p^{b|a}_{\beta_l\alpha_k}}.$$  

(28)

Hence, to obtain Born’s rule for the $b$-variable which matches the interference formula of total probability, we have to find $\lambda$ satisfying equations (27) and (28) and put such $\lambda$ into equations (25), then solve this system of equations. In general, it is a complex problem. Thus, finally, we can write the complete system of equations:

$$\sum_{l=1}^{3} \lambda_{l,jk} \sqrt{p_{\beta_l\alpha_j} p_{\beta_k\alpha_k}} = 0, \; j, k = 1, 2, 3; \quad (29)$$

$$p^{b}_{\beta_l} = p^{a}_{\alpha_i} p^{b|a}_{\beta_l\alpha_i} + p^{a}_{\alpha_i} p^{b|a}_{\beta_j\alpha_j} + p^{a}_{\alpha_i} p^{b|a}_{\beta_k\alpha_k} + 2\lambda_{l,ij} \sqrt{p_{\alpha_j} p^{a}_{\alpha_j} p^{b|a}_{\beta_l\alpha_i} p^{b|a}_{\beta_l\alpha_j}};$$

$$+ 2\lambda_{l,jk} \sqrt{p^{a}_{\alpha_j} p^{a}_{\alpha_k} p^{b|a}_{\beta_l\alpha_j} p^{b|a}_{\beta_l\alpha_k}}.$$  

(30)

$$\cos(\varphi_{\beta_m\alpha_1} - \varphi_{\beta_m\alpha_2}) = \lambda_{m,12};$$

$$\cos(\varphi_{\beta_m\alpha_2} - \varphi_{\beta_m\alpha_3}) = \lambda_{m,23};$$

$$\cos(\varphi_{\beta_m\alpha_1} - \varphi_{\beta_m\alpha_3}) = \lambda_{m,13}. \quad (31)$$

Solution of this system will provide us a complex probability amplitude $\psi$ such that $|\langle \psi | e^{b}_{\beta_l} \rangle|^2 = p^{b}_{\beta_l}$.

Let us consider the case of maximally unbiased matrix of transition probabilities,

$$p^{a|b}_{\beta_l\alpha_j} = 1/3, \; \forall \; l, j = 1, 2, 3$$  

(34)

Moreover, to simplify the task by the factor of three, we will put all

$$p^{b}_{\beta_l} = 1/3, \; \forall \; l = 1, 2, 3$$  

(35)

Now, let us introduce new variables $x > 0$ and $y > 0$:

$$\sqrt{p^{a}_{\alpha_i}/p^{a}_{\alpha_1}} = x, \; \sqrt{p^{a}_{\alpha_i}/p^{a}_{\alpha_3}} = y, \; \sqrt{p^{a}_{\alpha_2}/p^{a}_{\alpha_3}} = y/x, \quad (36)$$
That means that
\[ p_{\alpha_1}^a = \frac{x^2 y^2}{y^2 x^2 + x^2 + y^2}, \quad p_{\alpha_2}^a = \frac{y^2}{y^2 x^2 + x^2 + y^2}, \quad p_{\alpha_3}^a = \frac{x^2}{y^2 x^2 + x^2 + y^2}, \] (37)
and the condition \( p_{\alpha_1}^a + p_{\alpha_2}^a + p_{\alpha_3}^a = 1 \) always holds. Let proceed for a particular choice of interference coefficients (ansatz) \( \lambda_{l,12} = \mu, \quad \lambda_{l,13} = -\mu \), thus \( \lambda_{l,23} = 1 - 2\mu^2 \) by
\[ \lambda_{l,23} = \lambda_{l,12}\lambda_{l,13} \pm \sqrt{(1 - \lambda_{l,12}^2)(1 - \lambda_{l,13}^2)}. \] (38)

The system of equations (23) for \( \lambda \) under conditions (34) and (35) have the form:
\[ \lambda_{l,12} = \frac{3}{2} \left( x + \frac{1}{x} \right) \left( p_{\beta_{l}\alpha_1}^{b/a} \lambda_{l,12} - \frac{1}{3} \right), \]
\[ \lambda_{l,13} = \frac{3}{2} \left( y + \frac{1}{y} \right) \left( p_{\beta_{l}\alpha_1}^{b/a} \lambda_{l,13} - \frac{1}{3} \right), \]
\[ \lambda_{l,23} = \frac{3}{2} \left( \frac{y}{x} + \frac{x}{y} \right) \left( p_{\beta_{l}\alpha_1}^{b/a} \lambda_{l,13} - \frac{1}{3} \right). \]

We write this as:
\[ p_{\beta_{l}\alpha_1}^{b/a} = \frac{1}{3} + \frac{2\mu}{3(x + \frac{1}{x})}, \quad p_{\beta_{l}\alpha_1}^{b/a} = \frac{1}{3} - \frac{2\mu}{3(y + \frac{1}{y})}, \quad p_{\beta_{l}\alpha_3}^{b/a} = \frac{1}{3} + \frac{2(1 - 2\mu^2)}{3(x + \frac{1}{x})}, \] (40)
where \( \mu \) is a parameter. The probabilities given by (40) satisfy the relation in (19) which in this case looks as:
\[ (x^2 + 1)y^2 p_{\beta_{l}\alpha_1}^{b/a} + (y^2 + 1)x^2 p_{\beta_{l}\alpha_1}^{b/a} + (x^2 + y^2)p_{\beta_{l}\alpha_2}^{b/a} = \frac{2}{3}(x^2 y^2 + y^2 + x^2) \] (41)

**FIGURE 1.** Left side of the expression (44) is plotted versus parameter \( v = \lambda_{l,13} \) for the fixed values of \( x = 6.925, y = 2.225 \) and different signs before square roots in expressions for \( \lambda_{l,23} \), see (45), and \( \lambda_{l,12} \), see (46). We can see, that we can often satisfy (44) by properly choosing signs in expressions (45) and (46).
Putting (40) into (41), we get an equation for $\mu$:

$$2\mu^2 + (x - y)\mu - 1 = 0 \Rightarrow \mu = \frac{(y - x) \pm \sqrt{(x - y)^2 + 8}}{4} \tag{42}$$

We are interested in the case then all absolute value of lambdas are less than one

$$|\lambda_{l,ij}| < 1, \forall \ ij = 12, 13, 23. \tag{43}$$

It is satisfied when $|\mu| < 1$. So, for the case of $|x - y| < 1$, for both roots of (42) conditions $|\lambda_{l,ij}| < 1$ are valid, if $x - y > 1$, then (42) with the plus sign suits $|\lambda_{l,ij}| < 1$, otherwise $y - x > 1$, then (42) with the minus sign is valid.

Now we proceed in a general case, i.e. without ansatz $\lambda_{l,12} = \mu$, $\lambda_{l,13} = -\mu$, $\lambda_{l,23} = 1 - 2\mu^2$. Conditions (34) and (35) equation (19), which is equivalent to Born’s rule, comes down to:

$$y\lambda_{l,12} + x\lambda_{l,13} + \lambda_{l,23} = 0 \tag{44}$$

We should combine it with the constraint, see (38) for $\lambda_{l,12}, \lambda_{l,13}, \lambda_{l,23}$ to have simultaneous solution

$$\lambda_{l,23} = \lambda_{l,12}\lambda_{l,13} \pm \sqrt{(1 - \lambda_{l,12}^2)(1 - \lambda_{l,13}^2)} \tag{45}$$

We have two equations for three variables, thus we can express the solution as a one-parametric family. Let us choose $v = \lambda_{l,13}$ as a parameter. Then

$$\lambda_{l,12} = \frac{-xv(y + v) \pm \sqrt{(v^2 - 1)(v^2x^2 - y^2 - 1 - 2yv)}}{y^2 + 1 + 2yv} \tag{46}$$
and $\lambda_{l,23}$ can be obtained from equation (38). We have to make sure that $\lambda_{l,12}, \lambda_{l,13}, \lambda_{l,23}$ exist, are real and satisfy (43), given real and positive $x$ and $y$. In this, more general, case

$$
P_{b|a}^{\beta_1 \alpha_1} = \frac{1}{3} + \frac{2\lambda_{l,12}}{3(x + \frac{1}{x})}, \quad P_{b|a}^{\beta_1 \alpha_1} = \frac{1}{3} + \frac{2\lambda_{l,13}}{3(y + \frac{1}{y})}, \quad P_{b|a}^{\beta_1 \alpha_1} = \frac{1}{3} + \frac{2\lambda_{l,23}}{3\left(\frac{x}{y} + \frac{y}{x}\right)}, \quad (47)$$

Seeing that all values in the parentheses in (40) are greater than 2, each of the this probabilities non-negative and smaller than 2/3, if $\lambda_{l,12}, \lambda_{l,13}, \lambda_{l,23}$ are smaller than 1. The main problem is to describe possible rangers of parameters in (46) which give us $|\lambda_{l,ij}| \leq 1$, see figure 1–3. We remark that $\lambda_{l,13}$ is a a parameter, $|v| \leq 1$.

ACKNOWLEDGMENTS

One of the authors, Irina Basieva, is supported by Swedish Institute, post–doc fellowship. We would like to thank professor Andrei Khrennikov for fruitful discussions about quantum formalism and probability theory.

REFERENCES

Paper IV

Quantum-like representation algorithm for trichotomous observables.

Peter Nyman and Irina Basieva.

Quantum-like representation algorithm for trichotomous observables

Peter Nyman and Irina Basieva

International Center for Mathematical Modelling
in Physics and Cognitive Sciences,
Linnaeus University, S-35195, Sweden

August 17, 2011

Abstract

We study the problem of representing statistical data (of any origin) by a complex probability amplitude. This paper is devoted to representation of data collected from measurements of two trichotomous observables. The complexity of the problem eventually increases compared to the case of dichotomous observables. We see that only special statistical data (satisfying a number of nonlinear constraints) have the quantum–like representation.

1 Introduction

Quantum mechanics is based on a special probabilistic model in which probabilities are represented as squares of complex amplitudes. The question whether probabilistic data collected in some experiment can be described by the quantum probabilistic model is very complicated, see, e.g., [1–8,14,15]: given an experiment and collected data, can it be represented by complex probability amplitudes? Suppose one guess that the quantum formalism can be applied only to some micro-phenomena, but in the general case another model has to be used. Suppose one questioned the basic postulate of quantum mechanics that probabilistic data can be expressed through complex probability amplitudes. (This is not about comparison between classical and quantum physics. One speculates that micro-phenomena are even more exotic than in the conventional quantum model.) In this situation it is useful to have mathematical conditions which imply that the probabilistic data can be represented by using complex amplitudes and Born’s rule. This range of problems has been considered e.g. by R. Sorkin, see, e.g., [16], who speculated that the quantum model might be violated in measurements of incompatible trichotomous observables. He proposed a statistical test to check “quantumness of data” (later we will discuss it in more details). His test is related to the triple slit interference experiment. This experiment was recently performed by the research team of G. Weihs [16]. The data from this experiment passed Sorkin’s test. Hence, the hypothesis that the conventional quantum formalism describes not only the two slit experiment, but even the triple slit experiment could not be rejected on the basis of this test.

Recently C. Ududec, H. Barnum, J. Emerson [17] generalized Sorkin’s approach; they studied a huge class of possible generalizations of the conventional quantum model. (We state again that all models under consideration are nonclassical.) Their models violate Born’s rule for incompatible trichotomous observables.

In this paper we study the problem of describing experimental data by the probabilistic formalism of quantum mechanics in more details. We found necessary and sufficient conditions for existence of the complex amplitude representation of probabilities and a possibility to apply

*Research fellowship of Swedish Institute.
Born’s rule. As in Sorkin’s considerations, we study the case of the trichotomous observables. (The general problem was discussed in very detail in [18].)

Another reason to create the quantum-like (QL) representation of data is not so straightforward as the first one. In [19] A. Khrennikov presented a hypothesis that biological systems might use complex probabilistic amplitudes (“mental wave functions”) in processing of statistical data. If this hypothesis is correct, then these amplitudes can be reconstructed on the basis of collected experimental data. In psychology this approach got the name “constructive wave function approach”. By showing that data from, e.g., psychological experiments can be expressed using complex probability amplitudes one justifies a possibility to use the mathematical formalism of quantum mechanics to describe some psychological phenomena.

A general QL-representation algorithm (QLRA) was presented in [18]. This algorithm is based on the formula of total probability with interference term – a disturbance of the standard formula of total probability. Starting with experimental probabilistic data, QLRA produces a complex description of the class of probabilistic data which can be transferred into QL-amplitudes (the domain of application of QLRA) depends very much on the dimension. In [20] the simplest case of data generated by dichotomous observables was studied. In this paper we study trichotomous observables. The complexity of the problem increases incredibly compared to the two dimensional case.

2 Trichotomous incompatible quantum observables

2.1 Probabilities

Let \( \hat{a} \) and \( \hat{b} \) be two self-adjoint operators in three dimensional complex Hilbert space representing two trichotomous incompatible observables \( a \) and \( b \). They take values \( a = \alpha_i, \ i = 1, 2, 3 \) and \( b = \beta_j, l = 1, 2, 3 \) – spectra of operators. We assume that the operators have nondegenerate spectra, i.e., \( \alpha_i \neq \alpha_j, \beta_i \neq \beta_j, i \neq j \). Consider corresponding eigenvectors:

\[
\hat{a}e_{\alpha_i}^a = \alpha_ie_{\alpha_i}^a, \quad \hat{b}e_{\beta_j}^b = \beta_j e_{\beta_j}^b.
\]

Denote by \( \hat{P}_{\alpha_i}^a = |\langle e_{\alpha_i}^a | \rangle \langle e_{\alpha_i}^a | \rangle \) and \( \hat{P}_{\beta_j}^b = |\langle e_{\beta_j}^b | \rangle \langle e_{\beta_j}^b | \rangle \) one dimensional projection operators and by \( P_{\alpha_i}^a \) and \( P_{\beta_j}^b \) the observables repressed by there projections. Consider also projections

\[
\begin{align*}
\hat{P}_{\alpha_i}^a \perp & = \hat{P}_{\alpha_1}^a + \hat{P}_{\alpha_2}^a, \\
\hat{P}_{\alpha_2}^a & = \hat{P}_{\alpha_1}^a + \hat{P}_{\alpha_3}^a, \\
\hat{P}_{\alpha_3}^a & = \hat{P}_{\alpha_1}^a + \hat{P}_{\alpha_2}^a.
\end{align*}
\]

Here the observable \( P_{\alpha_i}^a = 1 \) if the result of the \( a \)-measurement is \( a = \alpha_i \) and \( P_{\alpha_i}^a = 0 \) if \( a \neq \alpha_i \). The observables \( P_{\beta_j}^b \) are defined in the same way. We have the following relation between events corresponding to measurements \( [P_{\alpha_i}^a = 0] = [P_{\alpha_2}^a = 1] \lor [P_{\alpha_3}^a = 1], [P_{\alpha_2}^a = 0] = [P_{\alpha_3}^a = 1] \lor [P_{\alpha_3}^a = 1], [P_{\alpha_3}^a = 0] = [P_{\alpha_1}^a = 1] \lor [P_{\alpha_2}^a = 1] \). There are given (by the QM-formalism) the probabilities

\[
\begin{align*}
& P_{\beta_j}^b = P_\psi(b = \beta) = ||\hat{P}_{\beta_j}^b \psi||^2 = |\langle \psi | e_{\beta_j}^b \rangle|^2, \\
& P_{\alpha_i}^a = P_\psi(a = \alpha) = ||\hat{P}_{\alpha_i}^a \psi||^2 = |\langle \psi | e_{\alpha_i}^a \rangle|^2.
\end{align*}
\]

There are also given (by the QM-formalism) conditional (transition) probabilities

\[
\begin{align*}
& P_{\beta_j}^b P_{\alpha_i}^a = P_\psi(b = \beta | P_{\alpha_i}^a = 1) = ||\hat{P}_\beta^b \hat{P}_\alpha^a \psi||^2 / ||\hat{P}_\alpha^a \psi||^2 = |\langle e_{\beta_j}^b | e_{\alpha_i}^a \rangle|^2.
\end{align*}
\]

We remark that non degeneration of the spectra implies that they do not depend on \( \psi \). Moreover, the matrix of transition probabilities is doubly stochastic. There are also given (\( \psi \)-depend)
probabilities

\[ p_{\beta_ia}\equiv P_\psi(b = \beta_i|P_{\alpha_k} = 1) = P_\psi(b = \beta_i|P_{\alpha_i} = 0) = \frac{||\hat{P}_a^b (\hat{P}_a^\alpha + \hat{P}_a^\alpha)\psi||^2}{||(\hat{P}_a^\alpha + \hat{P}_a^\alpha)\psi||^2}, \]

where \(ij, k \in \{1, 2, 3\}, j, k \neq i\). We have

\[ \frac{||\hat{P}_a^b (\hat{P}_a^\alpha + \hat{P}_a^\alpha)\psi||^2}{||P_{\alpha_j}^a + P_{\alpha_j}^a||^2} = \frac{||\langle b_\alpha| \langle b_\alpha|e_{\beta_\alpha}^a\rangle\langle b_\alpha|\psi\rangle + \langle b_\alpha|\langle b_\alpha|e_{\beta_\alpha}^a\rangle\langle b_\alpha|\psi\rangle\rangle ||^2}{||\langle b_\alpha|\langle b_\alpha|\psi\rangle + \langle b_\alpha|\langle b_\alpha|\psi\rangle\rangle ||^2} = \frac{||\langle b_\alpha|\langle b_\alpha|\psi\rangle + \langle b_\alpha|\langle b_\alpha|\psi\rangle\rangle^2}{||\langle b_\alpha|\langle b_\alpha|\psi\rangle + \langle b_\alpha|\langle b_\alpha|\psi\rangle\rangle^2 ||^2}. \]

Note that \( p_{\beta_ia\alpha_k} = p_{\beta_ia\alpha_k} \).

2.2 Probability amplitudes

Set \( \psi = (\psi|e_{\beta_\alpha}^b) \). Then by Born’s rule

\[ p_{\beta_\alpha}^b = |\psi_\beta|^2. \]  

(4)

We have

\[ \psi = \sum_{\beta_\alpha} \psi_{\beta_\alpha} e_{\beta_\alpha}^b. \]  

(5)

Thus these amplitudes give a possibility to reconstruct the state. We remark that \( \psi = \sum_{\alpha} \hat{P}_{\alpha}^a \psi \), and, hence,

\[ \psi_{\beta_\alpha} = \sum_{\alpha} \langle \hat{P}_{\alpha}^a \psi | e_{\beta_\alpha}^b \rangle. \]  

(6)

Each amplitude \( \psi_{\beta_\alpha} \) can be represented as the sum of three subamplitudes

\[ \psi_{\beta_\alpha} = \sum_{\alpha} \psi_{\beta_\alpha a} \]  

(7)

given by

\[ \psi_{\beta_\alpha a} = \langle \hat{P}_{\alpha}^a \psi | e_{\beta_\alpha}^b \rangle = (\psi|e_{\beta_\alpha}^a)\langle e_{\beta_\alpha}^a|e_{\beta_\alpha}^b \rangle. \]  

(8)

Hence, one can reconstruct the state \( \psi \) on the basis of nine amplitudes \( \psi_{\beta_\alpha a} \). We remark that \( |\psi_{\beta_\alpha a}|^2 = |(\psi|e_{\beta_\alpha}^a)\langle e_{\beta_\alpha}^a|\psi\rangle|^2 = p_{\beta_\alpha}^{b\alpha} P_{\beta_\alpha a\alpha_i}. \) In this notations

\[ p_{\beta_\alpha a\beta}^{b\alpha} = \frac{|\psi_{\beta_\alpha a}\psi_{\beta_\alpha a}|^2}{p_{\beta_\alpha a\alpha_i} + p_{\beta_\alpha a\alpha_i}^a}. \]  

(9)

Here \( |\psi_{\beta_\alpha a}| = \sqrt{p_{\beta_\alpha a\alpha_i}} \), and therefore

\[ \psi_{\beta_\alpha a} = \sqrt{p_{\beta_\alpha a\alpha_i}} e^{i\varphi_{\beta_\alpha a}}. \]

where \( \varphi_{\beta_\alpha a} = \text{arg} \psi_{\beta_\alpha a}. \) Moreover,

\[ \langle \psi|e_{\beta_\alpha}^a \rangle = \sqrt{p_{\beta_\alpha a\alpha_i}} e^{i\xi_{\alpha_i}}, \langle e_{\beta_\alpha}^a|\psi_{\beta_\alpha a} \rangle = \sqrt{p_{\beta_\alpha a\alpha_m}} e^{i\beta_{\alpha a\alpha_m}}. \]  

(10)

Hence,

\[ \varphi_{\beta_\alpha a\alpha_m} = \theta_{\beta_\alpha a} + \xi_{\alpha_i}. \]  

(11)
We have a system of equations for phases $\psi_{\beta l i}$ for $i, j, k, l \in \{1, 2, 3\}$,

$$|\psi_{\beta l i} + \psi_{\beta l j}|^2 = |\sqrt{p^a_{\alpha i} p^b_{\alpha j} e^{i\phi_{\beta l i}}} + \sqrt{p^a_{\alpha j} p^b_{\alpha i} e^{i\phi_{\beta l j}}}|^2 \quad (12)$$

$$= p^a_{\alpha i} p^b_{\alpha j} + p^a_{\alpha j} p^b_{\alpha i} + 2 \cos(\varphi_{\beta l i} - \varphi_{\beta l j}) \sqrt{p^a_{\alpha i} p^b_{\alpha j} p^b_{\alpha i} p^a_{\alpha j}}.$$  

We set

$$\lambda_{l,ij} \equiv \cos(\varphi_{\beta l i} - \varphi_{\beta l j}) \quad (13)$$

and we have

$$\lambda_{l,ij} = \frac{(p^a_{\alpha i} + p^a_{\alpha j}) p^b_{\alpha i} p^b_{\alpha j} - (p^a_{\alpha i} p^b_{\alpha j} + p^a_{\alpha j} p^b_{\alpha i})}{2 \sqrt{p^a_{\alpha i} p^b_{\alpha i} p^b_{\alpha j} p^a_{\alpha j}}}.$$  

We call $\lambda_{l,ij}$ for the coefficients of interference.

### 2.3 Formula of total probability with interference term

By using the definition of the amplitude $\psi_{\beta l i} = \langle \psi | e^{\alpha i} \rangle \langle e^{\alpha i} | e^{b j} \rangle$ we obtain

$$p^b_{\beta l} = |\psi_{\beta l i} + \psi_{\beta l j} + \psi_{\beta l k} + \psi_{\beta l l}|^2 = |\langle \psi | e^{\alpha i} \rangle \langle e^{\alpha i} | e^{b j} \rangle + \langle \psi | e^{\alpha i} \rangle \langle e^{\alpha j} | e^{b j} \rangle + \langle \psi | e^{\alpha i} \rangle \langle e^{\alpha j} | e^{b j} \rangle|^2 +$$

$$+ (\langle \psi | e^{\alpha i} \rangle \langle e^{\alpha j} | e^{b j} \rangle \langle e^{b j} | e^{b j} \rangle | \psi \rangle + \langle \psi | e^{\alpha i} \rangle \langle e^{\alpha j} | e^{b j} \rangle \langle e^{b j} | e^{b j} \rangle | \psi \rangle + \langle \psi | e^{\alpha i} \rangle \langle e^{\alpha j} | e^{b j} \rangle \langle e^{b j} | e^{b j} \rangle | \psi \rangle + \langle \psi | e^{\alpha j} \rangle \langle e^{\alpha j} | e^{b j} \rangle \langle e^{b j} | e^{b j} \rangle | \psi \rangle +$$

Finally, we obtain

$$p^b_{\beta l} = p^a_{\alpha i} p^b_{\alpha j} + p^a_{\alpha j} p^b_{\alpha i} + p^a_{\alpha k} p^b_{\alpha k} + 2 \cos(\varphi_{\beta l i} - \varphi_{\beta l j}) \sqrt{p^a_{\alpha i} p^b_{\alpha j} p^b_{\alpha i} p^a_{\alpha j}} + 2 \cos(\varphi_{\beta l i} - \varphi_{\beta l k}) \sqrt{p^a_{\alpha i} p^b_{\alpha k} p^b_{\alpha i} p^a_{\alpha k}}.$$  

This is nothing else than the formula of total probability with the interference term. It can be considered [22] as a perturbation of the classical formula of total probability

$$p^b_{\beta l} = p^a_{\alpha i} p^b_{\alpha i} + p^a_{\alpha j} p^b_{\alpha j} + p^a_{\alpha k} p^b_{\alpha k}.$$  

If all coefficients of interferes $\lambda_{l,ij} = 0$, then (16) coincides with (17).
2.4 Sorkin’s equality in conditional probabilistic form

We will derive Sorkin’s equality by putting (13) in (16),

\[
p^b_{\beta_1} = p^a_{\alpha_1} b_{a_{\alpha_1}} + p^a_{\alpha_2} b_{a_{\alpha_2}} + p^a_{\alpha_3} b_{a_{\alpha_3}} + \left( p^a_{\alpha_1} + p^a_{\alpha_2} \right) p^b_{\delta_1 a_{\alpha_1}} - \left( p^a_{\alpha_1} b_{a_{\alpha_1}} + p^a_{\alpha_2} b_{a_{\alpha_2}} \right)
\]

\[
+ \left( p^a_{\alpha_2} + p^a_{\alpha_3} \right) p^b_{\delta_1 a_{\alpha_2}} - \left( p^a_{\alpha_2} b_{a_{\alpha_2}} + p^a_{\alpha_3} b_{a_{\alpha_3}} \right)
\]

\[
+ \left( p^a_{\alpha_3} + p^a_{\alpha_1} \right) p^b_{\delta_1 a_{\alpha_3}} - \left( p^a_{\alpha_3} b_{a_{\alpha_3}} + p^a_{\alpha_1} b_{a_{\alpha_1}} \right)
\]

\[
= p^a_{\alpha_1} p^b_{\delta_1 a_{\alpha_1}} + \left( p^a_{\alpha_1} + p^a_{\alpha_2} \right) p^b_{\delta_1 a_{\alpha_1}} + \left( p^a_{\alpha_2} + p^a_{\alpha_3} \right) p^b_{\delta_1 a_{\alpha_2}} + \left( p^a_{\alpha_3} + p^a_{\alpha_1} \right) p^b_{\delta_1 a_{\alpha_3}}
\]

\[
- \left( p^a_{\alpha_1} b_{a_{\alpha_1}} + p^a_{\alpha_2} b_{a_{\alpha_2}} + p^a_{\alpha_3} b_{a_{\alpha_3}} \right)
\]

\[
= p^a_{\alpha_1} p^b_{\delta_1 a_{\alpha_1}} + p^a_{\alpha_2} p^b_{\delta_1 a_{\alpha_2}} + p^a_{\alpha_3} p^b_{\delta_1 a_{\alpha_3}} - p^a_{\alpha_1} b_{a_{\alpha_1}} - p^a_{\alpha_2} b_{a_{\alpha_2}} - p^a_{\alpha_3} b_{a_{\alpha_3}}
\]

This gives us the following constraint on the probabilities

\[
p^b_{\beta_1} = p^a_{\alpha_1} b_{a_{\alpha_1}} + p^a_{\alpha_2} b_{a_{\alpha_2}} + p^a_{\alpha_3} b_{a_{\alpha_3}}
\]

\[
+ p^a_{\alpha_1} p^b_{\delta_1 a_{\alpha_1}} + p^a_{\alpha_2} p^b_{\delta_1 a_{\alpha_2}} + p^a_{\alpha_3} p^b_{\delta_1 a_{\alpha_3}} - p^a_{\alpha_1} b_{a_{\alpha_1}} - p^a_{\alpha_2} b_{a_{\alpha_2}} - p^a_{\alpha_3} b_{a_{\alpha_3}}
\]

This equation coupling various quantum probabilities can be considered as encrypting of Born’s rule by using the language of probabilities. This is the discrete version of famous Sorkin equality [23,24].

2.5 Unitarity of transition operator

We now remark that bases consisting of $a$- and $b$-eigenvectors are orthogonal; hence the operator of transition from one basis to another is unitary. We can always select the $b$-basis in the canonical way

\[
e^{\theta}_{\beta_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad e^b_{\beta_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad e^b_{\beta_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]

(20)

In this system of coordinates the $a$-basis has the form

\[
e^{a}_{\alpha_1} = \begin{pmatrix} \sqrt{p_{\alpha_1 \alpha_1}} e^{i \theta_{\alpha_1 \alpha_1}} \\ \sqrt{p_{\alpha_1 \alpha_2}} e^{i \theta_{\alpha_1 \alpha_2}} \\ \sqrt{p_{\alpha_1 \alpha_3}} e^{i \theta_{\alpha_1 \alpha_3}} \end{pmatrix}, \quad e^a_{\alpha_2} = \begin{pmatrix} \sqrt{p_{\alpha_2 \alpha_1}} e^{i \theta_{\alpha_2 \alpha_1}} \\ \sqrt{p_{\alpha_2 \alpha_2}} e^{i \theta_{\alpha_2 \alpha_2}} \\ \sqrt{p_{\alpha_2 \alpha_3}} e^{i \theta_{\alpha_2 \alpha_3}} \end{pmatrix}, \quad e^a_{\alpha_3} = \begin{pmatrix} \sqrt{p_{\alpha_3 \alpha_1}} e^{i \theta_{\alpha_3 \alpha_1}} \\ \sqrt{p_{\alpha_3 \alpha_2}} e^{i \theta_{\alpha_3 \alpha_2}} \\ \sqrt{p_{\alpha_3 \alpha_3}} e^{i \theta_{\alpha_3 \alpha_3}} \end{pmatrix}
\]

(21)

The matrix

\[
U = \begin{pmatrix} \sqrt{p_{\alpha_1 \alpha_1}} e^{i \theta_{\alpha_1 \alpha_1}} & \sqrt{p_{\alpha_1 \alpha_2}} e^{i \theta_{\alpha_1 \alpha_2}} & \sqrt{p_{\alpha_1 \alpha_3}} e^{i \theta_{\alpha_1 \alpha_3}} \\ \sqrt{p_{\alpha_2 \alpha_1}} e^{i \theta_{\alpha_2 \alpha_1}} & \sqrt{p_{\alpha_2 \alpha_2}} e^{i \theta_{\alpha_2 \alpha_2}} & \sqrt{p_{\alpha_2 \alpha_3}} e^{i \theta_{\alpha_2 \alpha_3}} \\ \sqrt{p_{\alpha_3 \alpha_1}} e^{i \theta_{\alpha_3 \alpha_1}} & \sqrt{p_{\alpha_3 \alpha_2}} e^{i \theta_{\alpha_3 \alpha_2}} & \sqrt{p_{\alpha_3 \alpha_3}} e^{i \theta_{\alpha_3 \alpha_3}} \end{pmatrix}
\]

is unitary. Hence, we have the system of equations

\[
\sum_m \sqrt{p_{\beta_1 \alpha_1} p_{\beta_1 \alpha_1}} e^{i (\theta_{\beta_1 \alpha_1} - \theta_{\beta_1 \alpha_1})} = 0
\]

or

\[
\sum_m \sqrt{p_{\beta_1 \alpha_1} p_{\beta_1 \alpha_1}} \cos(\theta_{\beta_1 \alpha_1} - \theta_{\beta_1 \alpha_1}) = 0,
\]

\[
\sum_m \sqrt{p_{\beta_1 \alpha_1} p_{\beta_1 \alpha_1}} \sin(\theta_{\beta_1 \alpha_1} - \theta_{\beta_1 \alpha_1}) = 0,
\]

where ($i \neq j$) (For $i = j$, the unitarity condition is equivalent to normalization of the sum of probabilities by one) +. We now recall that the phases of the basis vectors $e^a_{\alpha}$ are coupled with
the phases of the amplitudes \( \psi_{\beta \alpha} \) by (11). Hence, we obtain a system of constraints on the later phases
\[
\sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \cos[(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) + (\xi_{\beta j} - \xi_{\alpha i})] = 0,
\]
(25)
\[
\sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \sin[(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) + (\xi_{\beta j} - \xi_{\alpha i})] = 0.
\]
(26)
Thus
\[
\cos(\xi_{\beta j} - \xi_{\alpha i}) \sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \cos(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) = 0,
\]
(27)
\[
- \sin(\xi_{\beta j} - \xi_{\alpha i}) \sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \sin(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) = 0;
\]
(28)
\[
\cos(\xi_{\beta j} - \xi_{\alpha i}) \sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \sin(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) = 0.
\]
(29)
Suppose now the following equalities hold
\[
\sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \sin(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) = 0,
\]
(29)
\[
\sum \sqrt{p_{\beta \alpha} p_{\beta \alpha}} \cos(\phi_{\beta m} \alpha_i - \phi_{\beta m} \alpha_j) = 0.
\]
(30)
Then by (27), (28) the equalities (23), (24) and, hence, (22) hold. Thus conditions (29), (30) imply unitarity of \( U \). It is clear that in turn (23), (24) imply (29), (30) for arbitrary \( \xi_i \). Thus the later conditions are equivalent to unitarity of \( U \).

3 Mutually unbiased bases

In previous considerations we introduced the coefficients of interference \( \lambda_{i,j} \) on the basis of the phases \( \varphi_{\beta \alpha} \), \( \varphi_{\beta \alpha} \) by \( \lambda_{i,j} = \cos(\varphi_{\beta \alpha} - \varphi_{\beta \alpha}) \). However, we see that they also could be defined by using only probabilistic data, see (14). Can we the go other way around and to find phases \( \varphi_{\beta \alpha} \), \( \varphi_{\beta \alpha} \) on the basis of the interference coefficients given by (14)? We will study this general problem in section 5. Now we would like consider an example. To be sure that the problem has a solution, we start with data and the corresponding interference coefficients generated by QM. In section 5 we shall operate with statistical data an arbitrary origin. First we show that there exist probabilistic data such that (29) and (30) hold, therefore consider the following situation. Let \( p_{\beta \alpha} = 1/3 \) where \( i, j = 1, 2, 3 \). We will also put
\[
\theta_{\beta \alpha} = \theta_{\beta \alpha} = - \theta_{\beta \alpha} = - \theta_{\beta \alpha} = 2\pi/3,
\]
(31)
where all the other \( \theta_{\beta \alpha} = 0, \ i, j = 1, 2, 3 \). We insert this in the basis in (21) and obtain the orthonormal basis
\[
e^a_{\alpha} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ w \\ 1 \end{pmatrix}, \ e^a_{\alpha} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ w \\ m \end{pmatrix}, \ e^a_{\alpha} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ w \end{pmatrix},
\]
(32)
where \( w = \exp^{i2\pi/3} \). Bases (20) and (32) are mutually unbiased, i.e. \( |\{e^a_{\beta} | e^a_{\alpha} \}| = 1/3 \). Then let
\[
\psi = \frac{1}{\sqrt{3}}(e_{\beta 1} + e^{i\gamma_1} e_{\beta 2} + e^{i\gamma_2} e_{\beta 3}).
\]
Note that $|\psi|^2 = 1$. Then by equation (2) and after some calculations this gives that,

\[
p_{\alpha_1} = |\langle e_\alpha |\psi \rangle|^2 = \frac{1}{9} \left(3 - \cos (\gamma_1) - \cos (\gamma_1 - \gamma_2) + 2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2)\right),
\]

(33)

\[
p_{\alpha_2} = |\langle e_\alpha |\psi \rangle|^2 = \frac{1}{9} \left(3 - \cos (\gamma_1) + 2 \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) - \sqrt{3} \sin (\gamma_1) - \sqrt{3} \sin (\gamma_2)\right),
\]

(34)

\[
p_{\alpha_3} = |\langle e_\alpha |\psi \rangle|^2 = \frac{1}{9} \left(3 + 2 \cos (\gamma_1) - \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) - \sqrt{3} \sin (\gamma_1 - \gamma_2) + \sqrt{3} \sin (\gamma_2)\right).
\]

(35)

It is straightforward to see that $p_{\alpha_1} + p_{\alpha_2} + p_{\alpha_3} = 1$. The conditional probabilities\footnote{Recall that $p_{\beta|\alpha_{l,k}} = p_{\beta|\alpha_{k,l}}$.} $p_{\beta|\alpha_{l,k}}$ for $l, k, j = 1, 2, 3, \ k \neq j$ are calculated by (9)

\[
p_{\beta_1|\alpha_{12}} = -\frac{4 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - 2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + 2 \sqrt{3} \sin (\gamma_2) - 6}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - \sqrt{3} \sin (\gamma_2) + 6)}
\]

\[
p_{\beta_1|\alpha_{13}} = \frac{-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) + 2 \sqrt{3} \sin (\gamma_1) - \sqrt{3} \sin (\gamma_1 - \gamma_2) - 2 \cos (\gamma_2) - \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_2) + 3}{3 (-2 \cos (\gamma_1) - \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - \sqrt{3} \sin (\gamma_2) + 6)}
\]

(36)

\[
p_{\beta_1|\alpha_{23}} = \frac{-4 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - 2 \cos (\gamma_2) - \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + 2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + \sqrt{3} \sin (\gamma_2) + 6}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + \sqrt{3} \sin (\gamma_2) + 6)}
\]

\[
p_{\beta_2|\alpha_{12}} = \frac{-2 \cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) - 2 \sqrt{3} \sin (\gamma_1) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - 6}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + \sqrt{3} \sin (\gamma_2) + 6)}
\]

\[
p_{\beta_2|\alpha_{13}} = \frac{-4 \cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) - 2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + \sqrt{3} \sin (\gamma_2) + 6}{3 (-2 \cos (\gamma_1) - \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - \sqrt{3} \sin (\gamma_2) + 6)}
\]

\[
p_{\beta_2|\alpha_{23}} = \frac{-2 \cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) - \sqrt{3} \sin (\gamma_1 - \gamma_2) + 3}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - \sqrt{3} \sin (\gamma_2) + 3)}
\]

\[
p_{\beta_3|\alpha_{12}} = \frac{\cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) - 4 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + 6}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) - \sqrt{3} \sin (\gamma_1 - \gamma_2) - 3)}
\]

(37)

\[
p_{\beta_3|\alpha_{13}} = \frac{-2 \cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + 6}{3 (-2 \cos (\gamma_1) - \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1 - \gamma_2) - \sqrt{3} \sin (\gamma_2) + 6)}
\]

\[
p_{\beta_3|\alpha_{23}} = \frac{-2 \cos (\gamma_1) - 2 \cos (\gamma_1 - \gamma_2) + \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) + \sqrt{3} \sin (\gamma_1 - \gamma_2) + 6}{3 (-2 \cos (\gamma_1) + \cos (\gamma_1 - \gamma_2) - \cos (\gamma_2) + \sqrt{3} \sin (\gamma_1) - \sqrt{3} \sin (\gamma_1 - \gamma_2) + 3)}
\]

All probabilities are found. Moreover we find that $p_{\beta_k} = 1/3$, $l = 1, 2, 3$ by inserting (13) in (16) or directly by (19). We let $\gamma_1 = \gamma_2$ in order to get more compact expressions. This leads to

\[
p_{\beta_1|\alpha_{12}} = \frac{4 \sin (\gamma_2 + \frac{\pi}{6}) - 5}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}, \quad p_{\beta_1|\alpha_{13}} = \frac{2}{3}, \quad p_{\beta_1|\alpha_{23}} = \frac{4 \sin (\gamma_2 + \frac{\pi}{6}) - 5}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}
\]

\[
p_{\beta_2|\alpha_{12}} = \frac{-5 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_2) + 8}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}, \quad p_{\beta_2|\alpha_{23}} = \frac{1}{6}, \quad p_{\beta_2|\alpha_{23}} = \frac{2 (-2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_2) - 4)}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}
\]

\[
p_{\beta_3|\alpha_{12}} = \frac{2 (-2 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_2) - 4)}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}, \quad p_{\beta_3|\alpha_{23}} = \frac{1}{6}, \quad p_{\beta_3|\alpha_{23}} = \frac{-5 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_2) + 8}{3 (2 \sin (\gamma_2 + \frac{\pi}{6}) - 7)}
\]

Here we calculate $\lambda_{l,j,k}$ when $\gamma_2 = \gamma_1$

\[
\lambda_{1,12} = \lambda_{1,23} = -\frac{\sqrt{1 + \sin (\gamma_2 + \frac{\pi}{6})}}{\sqrt{10 - 8 \sin (\gamma_2 + \frac{\pi}{6})}}, \quad \lambda_{1,13} = 1
\]
\[ \lambda_{2,12} = \lambda_{3,23} = -\frac{4 \cos (\gamma_2) + 2\sqrt{3} \sin (\gamma_2) + 1}{2\sqrt{4 \sin \left(\frac{\pi}{6} - 2\gamma_2\right) + 2 \sin \left(\gamma_2 + \frac{\pi}{6}\right) + 6}}, \quad \lambda_{2,13} = \lambda_{3,12} = -\frac{1}{2} \]

\[ \lambda_{2,23} = \lambda_{3,12} = -\frac{-5 \cos (\gamma_2) + \sqrt{3} \sin (\gamma_2) - 1}{2\sqrt{4 \sin \left(\frac{\pi}{6} - 2\gamma_2\right) + 2 \sin \left(\gamma_2 + \frac{\pi}{6}\right) + 6}} \]

We calculate numerically the extreme values for all \( \lambda \) to prove the existence of angles between 0 and \( 2\pi \) for \( \arccos \lambda \). Thus, calculate \( \frac{d}{d\gamma_2} \lambda = 0 \) and the limits when \( \gamma_2 \rightarrow \pm 4\pi/3 \),

\[ \frac{d}{d\gamma_2} \lambda_{1,12} = \frac{d}{d\gamma_2} \lambda_{1,23} = 0 \Rightarrow \gamma_2 = \frac{\pi}{3} \pm 2\pi C, \; C \in \mathbb{Z} \]

and \( \lambda_{1,12} = \lambda_{1,23} = -1, \; \gamma_2 = \frac{\pi}{3} \pm 2\pi C \). The problems arise when \( \gamma_2 \rightarrow 4\pi/3 \) for \( \lambda_{1,12} \) and \( \lambda_{1,23} \), since \( \lim_{\gamma_2 \rightarrow \pm 4\pi/3} \). Let \( \lambda_{1,12} = \lambda_{1,23} = 0 \) when \( \gamma_2 = 4\pi/3 \). Moreover there exist angles

\[ \frac{\pi}{2} \leq \varphi_{\beta_1,\alpha_1} - \varphi_{\beta_1,\alpha_2} \leq \pi, \quad \text{(38)} \]
\[ \frac{\pi}{2} \leq \varphi_{\beta_1,\alpha_2} - \varphi_{\beta_1,\alpha_3} \leq \pi, \quad \text{(39)} \]

where \( \arccos \lambda_{1,12} = \varphi_{\beta_1,\alpha_1} - \varphi_{\beta_1,\alpha_2} \). We see that the denominator of \( \lambda_{2,23}, \lambda_{3,23}, \lambda_{3,12}, \lambda_{2,12} \) goes to zero when \( \gamma_2 \rightarrow 4\pi/3 \pm \).

We therefore examine the limits

\[ \lim_{\gamma_2 \rightarrow 4\pi/3 \pm} \lambda_{2,23} = \lim_{\gamma_2 \rightarrow 4\pi/3 \pm} \lambda_{2,12} = \lim_{\gamma_2 \rightarrow 4\pi/3 \pm} \lambda_{2,23} = \lim_{\gamma_2 \rightarrow 4\pi/3 \pm} \lambda_{2,12} = \pm \frac{\sqrt{3}}{2}. \quad \text{(40)} \]

Analysis of extreme values provides us with the following

\[ \frac{d}{d\gamma_2} \lambda_{2,23} = \frac{d}{d\gamma_2} \lambda_{3,12} = 0 \Rightarrow \gamma_2 = 2\pi C, \; C \in \mathbb{Z} \]

and

\[ \frac{d}{d\gamma_2} \lambda_{3,23} = \frac{d}{d\gamma_2} \lambda_{2,12} = 0 \Rightarrow \gamma_2 = \frac{2\pi}{3} \pm 2\pi C, \; C \in \mathbb{Z}. \quad \text{(41)} \]

The minimum values are given by (39), \( \min \lambda_{2,23} = \min \lambda_{3,23} = \min \lambda_{2,12} = \min \lambda_{3,12} = -\frac{\sqrt{3}}{2}, \; \gamma_2 \rightarrow 4\pi/3 \) or \( \gamma_2 \rightarrow -4\pi/3 \). The maximum values are given by (40) and (41), \( \max \lambda_{2,23} = \max \lambda_{3,23} = \max \lambda_{2,12} = 1, \; \gamma_2 = 2\pi C, \; C \in \mathbb{Z}. \; \max \lambda_{3,12} = \max \lambda_{3,12} = 1, \; \gamma_2 = \frac{2\pi}{3} \pm 2\pi C; C \in \mathbb{Z}. \) This prove that there exist angles such that

\[ 0 \leq \varphi_{\beta_2,\alpha_2} - \varphi_{\beta_1,\alpha_3} \leq \frac{5\pi}{6}, \quad \text{(42)} \]
\[ 0 \leq \varphi_{\beta_3,\alpha_2} - \varphi_{\beta_1,\alpha_3} \leq \frac{5\pi}{6}, \quad \text{(43)} \]
\[ 0 \leq \varphi_{\beta_3,\alpha_1} - \varphi_{\beta_2,\alpha_2} \leq \frac{5\pi}{6}, \quad \text{(43)} \]

and

\[ \varphi_{\beta_1,\alpha_1} - \varphi_{\beta_1,\alpha_3} = 0, \]
\[ \varphi_{\beta_2,\alpha_1} - \varphi_{\beta_2,\alpha_3} = \frac{2\pi}{3}, \]
\[ \varphi_{\beta_3,\alpha_1} - \varphi_{\beta_3,\alpha_2} = \frac{2\pi}{3} \]
4 Triple slit experiment

An interesting example of interplay of two incompatible trichotomous observables is given by the triple slit experiment – a natural generalization of the well know two slit experiment. There are given: a) a source of quantum systems which has very low intensity (so it might be interpreted as single-particle source); 2) a screen with three slits \( \alpha = 1, 2, 3 \) and each of them can be open or close on the demand; c) registration screen; typically it is covered by photo-emulsion; this produces the continuous interference picture; we shall consider discrete experiment. The \( a \)-observable gives slits’ which is so to say is passed by a particle on the way from the source to the registration screen. To measure \( a \), an experimenter puts three detectors directly behind slits. We set \( a = \alpha_i \), if the detector behind the \( i \)th slit produces a click. By the assumption the source has so low intensity that one can neglect by double clicks (e.g., the detectors never click simultaneously). We can find probabilities \( p_{\beta}^{a} \alpha, \alpha = 1, 2, 3 \). This is the first experiments producing \( a \)-probabilities. Now we consider basic experiments.

To make the second variable discrete, we put detectors in three fixed places of the registration slit. It gives us the observable \( b = \beta, \beta = 1, 2, 3 \). Thus \( \beta = 1 \) if the first detector clicks and so on.

The experiment is repeated at a few incompatible contexts, see [18] for general presentation

\[
C_{123} : \text{all slits are open; probabilities of } b \text{ detection collected in this context are probabilities } p_{\beta}^{b} = P_{\beta}(b = \beta) \text{ from section 2.1. In the QM-formalism context } C_{123} \text{ is represented by a quantum state } \psi.
\]

\[
C_{\alpha}, \alpha = 1, 2, 3 : \text{only the slit } \alpha \text{ is open; corresponding probabilities are } p_{\beta}^{b} \text{, transition probabilities. In the QM-formalism context } C_{\alpha} \text{ is represented by the quantum state } \psi_\alpha.
\]

\[
C_{\alpha_i, i \neq j} : \text{only slits } \alpha_i \text{ and } \alpha_j \text{ are open (so the slit } \alpha_k, k \neq i, j, \text{ is closed; corresponding probabilities are } p_{\beta}^{b} p_{\beta}^{b} \text{, transition probabilities. In the QM-formalism context } C_{\alpha_i, i \neq j} \text{ is represented by the quantum state } \psi_{\alpha_i, i \neq j} = \frac{(P_{\alpha_i} + P_{\alpha_j})\psi}{\|P_{\alpha_i} + P_{\alpha_j}\psi\|}.
\]

Thus all probabilities discussed in section 2.1 can be collected in this experiment. It is possible to check whether these experimental probabilities match the predictions of QM. The easiest way for experimenters is to check Sorkin equality (19).

Recently the group of Gregor Weihs performed the triple slit experiment\(^2\), see [16]. It is too early to make a definite conclusion on the results of Weihs’ experiment. It is not easy to separate the possible "nonquantum effect", violation of Sorkin’s equality, from the contribution induced by nonlinearity of detectors. However, preliminary analysis does not reject the conventional quantum model.

5 Construction of a complex probability amplitude satisfying Born’s rule

Now we have a pair of trichotomous observables \( a \) and \( b \) taking values \( a = \alpha_i, i = 1, 2, 3 \) and \( b = \beta_l, l = 1, 2, 3 \). We do not assume that they have any relation to quantum physics; e.g., these are some random variables observed in biology or finances. It is assumed that there are given probability distributions of these variables

\[
p_{\beta_l}^{b} = P(b = \beta_l), \quad p_{\alpha_i}^{a} = P(a = \alpha_i).
\]

Thus

\[
\sum_{l=1}^{3} p_{\beta_l}^{b} = 1, \quad \sum_{i=1}^{3} p_{\alpha_i}^{a} = 1. \quad (44)
\]

\(^2\)It is surprising that it has not been done for long ago!
It is also assumed that there are given conditional probabilities \( p_{b|a}^{\beta \alpha_j} = P(b = \beta_l|a = \alpha_i) \). We know that for any sort of data the matrix of transition probabilities is stochastic, i.e., for each \( \alpha_i \)

\[
\sum_{l=1}^{3} p_{b|a}^{\beta_l \alpha_i} = 1. 
\]

Finally, we assume a possibility to collect the data on measurements of observables \( P_{\alpha_i} \), \( i = 1, 2, 3 \), probabilities \( p_{b|a}^{\beta_1 \alpha_3} = P(b = \beta_l|P_{\alpha_i} = 0) \). “The detector corresponding to the value \( \alpha = \alpha_i \) does not click, so the value of \( \alpha \) is either \( \alpha = \alpha_j \) or \( \alpha = \alpha_k \), where \( j, k \neq i \). However, we do not know the value of \( \alpha \). In this context we measure the \( b \)-variable.” For any sort of data, we have

\[
\sum_{l=1}^{3} p_{b|a}^{\beta_l \alpha_{ij}} = 1. 
\]

5.1 Complex amplitude matching Born’s rule for one observable

Now we want to find a complex probability amplitude \( \psi_{\beta_l} \) such that Born’s rule (for the \( b \)-variable) holds: \( |\psi_{\beta_l}|^2 = p_{\beta_l}^{\beta_l} \). We copy the QM-scheme, so we represent \( \psi_{\beta_l} = \psi_{\beta_l}^{\beta_1} + \psi_{\beta_l}^{\beta_2} + \psi_{\beta_l}^{\beta_3} \), where the sub-amplitudes \( \psi_{\beta_l}^{\beta_1} = \sqrt{p_{\alpha_i}^{\beta_1} p_{\alpha_i}^{\beta_2}} e^{i \phi_{\beta_l}^{\alpha_i}} \), and phases are determined by the system of equations (13). It is convenient to work with the interference coefficients, see [18], given by right-hand sides of these equations

\[
\lambda_{i,j} = \lambda_{i,ji} = \frac{(p_{\alpha_i}^{\beta_1} + p_{\alpha_i}^{\beta_2}) p_{\alpha_i}^{\beta_1} - (p_{\alpha_i}^{\beta_1} p_{\alpha_i}^{\beta_2} + p_{\alpha_i}^{\beta_2} p_{\alpha_i}^{\beta_2})}{2 \sqrt{p_{\alpha_i}^{\beta_1} p_{\alpha_i}^{\beta_2} p_{\alpha_i}^{\beta_2} p_{\alpha_i}^{\beta_2}}}. 
\]

Interference coefficients obtained in quantum physics are always bounded by 1:

\[
|\lambda_{i,j}| \leq 1. 
\]

However, since we start with data of any origin, the condition (48) has to be checked to proceed to representation of data by complex amplitudes.\(^3\) If the system of equations, \( m = 1, 2, 3 \),

\[
\begin{align*}
\cos(\phi_{\beta_m \alpha_1} - \phi_{\beta_m \alpha_2}) &= \lambda_{m,12}, \\
\cos(\phi_{\beta_m \alpha_2} - \phi_{\beta_m \alpha_3}) &= \lambda_{m,23}, \\
\cos(\phi_{\beta_m \alpha_1} - \phi_{\beta_m \alpha_3}) &= \lambda_{m,13},
\end{align*}
\]

has a solution (three phases) then we can construct the probability amplitudes \( \psi_{\beta_l \alpha_i} \) and, hence, the probability amplitudes \( \psi_{\beta_l} \), and the corresponding vector \( \psi \).

However, in general such amplitudes will not provide a solution of the “inverse Born problem”, namely, Born’s rule can be violated. To obtain the real solution one should solve the system (49) under the constraint (19). Thus, to proceed toward a proper complex amplitude, one should first check the validity of (19) and then to solve the system (49). It is convenient to express “triple probabilities” \( p_{\beta_m \alpha_i} \) through coefficients of interference

\[
p_{\beta_m \alpha_1} = \frac{1}{p_{\alpha_i} + p_{\alpha_j}} (p_{\alpha_i} p_{\beta_m \alpha_i} + p_{\alpha_j} p_{\beta_m \alpha_j} - 2 \lambda_{\beta_m \alpha_1} \sqrt{p_{\alpha_i} p_{\beta_m \alpha_i} p_{\alpha_j} p_{\beta_m \alpha_j}}). 
\]

We remark that if (48) holds, then triple probabilities given by (50) are always nonnegative. By using the \( \lambda \)-variables normalization equations (46) can be written as \( (j, k = 1, 2, 3) \)

\[
\sum_{l=1}^{3} \lambda_{i,jk} \sqrt{p_{\beta_{m} \alpha_j} p_{\beta_{m} \alpha_k}} = 0. 
\]

\(^3\)If this condition is violated then data may be represented by so called hyperbolic probabilistic amplitudes [21].
We also can write Sorkin’s equality (in fact, the formula of total probability with interference terms) as

\[ p_{b}^{l} = p_{\alpha}^{a} b_{\beta l a}^{a} + p_{\alpha}^{a} b_{\beta l a}^{a} + p_{\lambda}^{a} b_{\beta l a}^{a} + 2\lambda_{l,j}^{b} \sqrt{p_{a}^{a} p_{a}^{a} p_{a}^{a} p_{a}^{a}} b_{\beta l a}^{a} \]

\[ + 2\lambda_{l,j,k}^{b} \sqrt{p_{a}^{a} p_{a}^{a} p_{a}^{a} p_{a}^{a}} b_{\beta l a}^{a} \]

Hence, to obtain Born’s rule for the \( b \)-variable which matches the interference formula of total probability, we have to find \( \lambda \) satisfying equations (51) and (52) and put such \( \lambda \) into equations (49), then solve this system of equations. In general, it is a complex problem.

Thus, finally, we can write the complete system of equations:

\[ \sum_{l=1}^{3} \lambda_{l,j,k}^{b} \sqrt{p_{\beta l a}^{a} p_{\beta l a}^{a}} = 0, \quad j, k = 1, 2, 3; \]  

\[ p_{b}^{l} = p_{\alpha}^{a} b_{\beta l a}^{a} + p_{\alpha}^{a} b_{\beta l a}^{a} + p_{\lambda}^{a} b_{\beta l a}^{a} + 2\lambda_{l,j}^{b} \sqrt{p_{a}^{a} p_{a}^{a} p_{a}^{a} p_{a}^{a}} b_{\beta l a}^{a} \]

\[ + 2\lambda_{l,j,k}^{b} \sqrt{p_{a}^{a} p_{a}^{a} p_{a}^{a} p_{a}^{a}} b_{\beta l a}^{a} \]

\[ \cos(\phi_{\beta l a} - \phi_{\beta l a}) = \lambda_{m,12} \]

\[ \cos(\phi_{\beta l a} - \phi_{\beta l a}) = \lambda_{m,23} \]

\[ \cos(\phi_{\beta l a} - \phi_{\beta l a}) = \lambda_{m,13} \]

Solution of this system will provide us a complex probability amplitude \( \psi \) such that \( |\langle \psi | e^{i\theta} \rangle|^{2} = p_{b}^{l} \)

Let us consider the case of maximally unbiased matrix of transition probabilities:

\[ p_{b}^{l} = 1/3, \quad \forall l, j = 1, 2, 3 \]  

Moreover, to simplify the task by the factor of three, we will put all

\[ p_{b}^{l} = 1/3, \quad \forall l = 1, 2, 3 \]

Now, let us introduce new variables \( x > 0 \) and \( y > 0 \):

\[ \sqrt{p_{a}^{a}/p_{a}^{a}} = x, \quad \sqrt{p_{b}^{a}/p_{b}^{a}} = y, \quad \sqrt{p_{a}^{a}/p_{a}^{a}} = y/x, \]

That means that

\[ p_{a}^{a} = \frac{x^{2}y^{2}}{y^{2}x^{2} + x^{2} + y^{2}}, \quad p_{a}^{a} = \frac{y^{2}}{y^{2}x^{2} + x^{2} + y^{2}}, \quad p_{a}^{a} = \frac{x^{2}}{y^{2}x^{2} + x^{2} + y^{2}} \]

and the condition \( p_{a}^{a} + p_{a}^{a} + p_{a}^{a} = 1 \) always holds. Let proceed for a particular choice of interference coefficients (ansatz) \( \lambda_{l,12} = \mu, \quad \lambda_{l,13} = -\mu, \) thus \( \lambda_{l,23} = 1 - 2\mu^{2} \) by

\[ \lambda_{l,23} = \lambda_{l,12}\lambda_{l,13} \pm \sqrt{(1-\lambda_{l,12}^{2})(1-\lambda_{l,13}^{2})} \]

The system of equations (47) for \( \lambda \) under conditions (58) and (59) have the form:

\[ \lambda_{l,12} = \frac{3}{2} \left( x + \frac{1}{x} \right) \left( \frac{b_{\beta l a}^{a}}{p_{\beta l a}^{a} p_{\beta l a}^{b}} - \frac{1}{3} \right) \]

\[ \lambda_{l,13} = \frac{3}{2} \left( y + \frac{1}{y} \right) \left( \frac{b_{\beta l a}^{a}}{p_{\beta l a}^{a} p_{\beta l a}^{b}} - \frac{1}{3} \right) \]

\[ \lambda_{l,23} = \frac{3}{2} \left( y + \frac{1}{y} \right) \left( \frac{b_{\beta l a}^{a}}{p_{\beta l a}^{a} p_{\beta l a}^{b}} - \frac{1}{3} \right) \]
We write this as:

\[ p_{b|a|\alpha_{12}} = \frac{1}{3} + \frac{2\mu}{3(x + \frac{1}{2})}, \quad p_{b|a|\alpha_{13}} = \frac{1}{3} - \frac{2\mu}{3(y + \frac{1}{2})}, \quad p_{b|a|\alpha_{23}} = \frac{1}{3} + \frac{2(1 - 2\mu^2)}{3 \left( \frac{y}{y} + \frac{1}{y} \right)}, \]  

(64)

where \( \mu \) is a parameter. The probabilities given by (64) satisfy the relation in (19) which in this case looks as:

\[ (x^2 + 1)y^2p_{b|\alpha_{12}} + (y^2 + 1)x^2p_{b|\alpha_{13}} + (x^2 + y^2)p_{b|\alpha_{23}} = \frac{2}{3}(x^2y^2 + y^2 + x^2) \]  

(65)

Putting (64) into (65), we get an equation for \( \mu \):

\[ 2\mu^2 + (x - y)\mu - 1 = 0 \Rightarrow \mu = \frac{(y - x) \pm \sqrt{(x - y)^2 + 8}}{4} \]  

(66)

We are interested in the case then all absolute value of lambdas are less than one

\[ |\lambda_{i,j}| < 1, \forall ij = 12, 13, 23. \]  

(67)

It is satisfied when \( |\mu| < 1 \). So, for the case of \( |x - y| < 1 \), for both roots of (66) conditions \( |\lambda_{i,j}| < 1 \) are valid, if \( x - y > 1 \), then (66) with the plus sign suits \( |\lambda_{i,j}| < 1 \), otherwise \( y - x > 1 \), then (66) with the minus sign is valid.

Now we proceed in a general case, i.e. without ansatz \( \lambda_{1,12} = \mu, \lambda_{1,13} = -\mu, \lambda_{1,23} = 1 - 2\mu^2 \). Conditions (58) and (59) equation (19), which is equivalent to Born’s rule, comes down to:

\[ y\lambda_{1,12} + x\lambda_{1,13} + \lambda_{1,23} = 0 \]  

(68)

We should combine it with the constraint, see (62) for \( \lambda_{1,12}, \lambda_{1,13}, \lambda_{1,23} \) to have simultaneous solution

\[ \lambda_{1,23} = \lambda_{1,12}\lambda_{1,13} \pm \sqrt{(1 - \lambda_{1,12}^2)(1 - \lambda_{1,13}^2)} \]  

(69)

We have two equations for three variables, thus we can express the solution as a one-parametric family. Let us choose \( v = \lambda_{1,13} \) as a parameter. Then

\[ \lambda_{1,12} = -xv(y + v) \pm \sqrt{(v^2 - 1)(v^2x^2 - y^2 - 1 - 2yv)} \]  

\[ y^2 + 1 + 2yv \]  

(70)
and $\lambda_{l,23}$ can be obtained from equation (62). We have to make sure that $\lambda_{l,12}, \lambda_{l,13}, \lambda_{l,23}$ exist, are real and satisfy (67), given real and positive $x$ and $y$. In this, more general, case

$$
\begin{align*}
 p_{b|a}^{\beta\alpha_1} &= \frac{1}{3} + \frac{2\lambda_{l,12}}{3(x + \frac{1}{2})}, \\
 p_{b|a}^{\beta\alpha_1} &= \frac{1}{3} + \frac{2\lambda_{l,13}}{3(y + \frac{1}{2})}, \\
 p_{b|a}^{\beta\alpha_1} &= \frac{1}{3} + \frac{2\lambda_{l,23}}{3\left(\frac{2}{y} + \frac{2}{x}\right)},
\end{align*}
$$

(71)

Seeing that all values in the parentheses in (64) are greater than 2, each of the this probabilities non-negative and smaller than 2/3, if $\lambda_{l,12}, \lambda_{l,13}, \lambda_{l,23}$ are smaller than 1. The main problem is to describe possible ranges of parameters in (70) which give us $|\lambda_{l,ij}| \leq 1$, see figure 1–3. We remark that $\lambda_{l,13}$ is a parameter, $|v| \leq 1$.

![Figure 2: Left side of the expression (68) is plotted versus parameters $x, y$, for the fixed $v = \lambda_{l,13} = 0, 555$. Minus sign is taken before square root in (70), plus sign is taken in (69) when obtaining $\lambda_{l,13}$ values. We can see that it is equal to zero as (68) demands on a large scope of $x, y$ parameters values.](image)

5.2 Complex amplitude matching Born’s rule for two observables

We now want to select an orthonormal basis $e_{\alpha}^a$ such that, for the state $\psi$ constructed in the previous section, $|\langle \psi | e_{\alpha}^a \rangle|^2 = p_{a}^\alpha$. We turn to considerations of section 2.5. Since vectors of this basis can be selected up to $e^{i\xi_{\alpha}}$. We can select $\theta_{\beta\alpha_k} = \phi_{\beta\alpha_k}$, i.e., set $\xi_{\alpha} = 0$. Of course, to guarantee orthogonality of this basis, constraints (29), (30) should be taken into account:

$$
\begin{align*}
 \sum_m \sqrt{p_{b|\alpha_1} p_{b|\alpha_j}} \lambda_{m,ij} &= 0, \\
 \sum_m \epsilon_{m,ij} \sqrt{p_{b|\alpha_1} p_{b|\alpha_j}} \sqrt{1 - \lambda_{m,ij}^2} &= 0,
\end{align*}
$$

(72) (73)

where $\epsilon_{m,ij}$ are signs which are selected in a proper way.

Moreover, the matrix of transition probabilities has to be doubly stochastic, i.e., besides (45), we should have

$$
\sum_{l=j}^3 \frac{b_{l|a}}{p_{b|\alpha_j}} = 1
$$

(74)
Figure 3: Value of $\lambda_{l,12}$ as function (70) of parameters $x, y$, for the fixed $v = \lambda_{l,13} = 0.585$. Plus sign is taken before square root in 70). One may see that value $\lambda_{l,12}$ is within $(-1,1)$ limits for the most part of the values $x$ and $y$.

for each $l = 1, 2, 3$.

Under these conditions the complex amplitude $\psi$ produced by our algorithm matches with Born’s rule for both observables, $\alpha$ and $\beta$.

**Example 1.** We can take

$$\lambda_{1,12} = \mu, \lambda_{1,23} = -\mu, \lambda_{1,13} = 0;$$

$$\lambda_{2,12} = -\mu, \lambda_{2,23} = 0, \lambda_{2,13} = \mu;$$

$$\lambda_{3,12} = 0, \lambda_{3,23} = \mu, \lambda_{3,13} = -\mu.$$

This gives us the solution $\mu = -\frac{1}{\sqrt{2}}$. First take the plus-case(e.i. $\mu = \pm \frac{1}{\sqrt{2}}$). We select $\phi_{m1} = \nu_m, m = 1, 2, 3$. For $\beta_1$, we obtain the system of equations: $\cos(\nu_1 - \phi_{12}) = \frac{1}{\sqrt{2}}, \cos(\phi_{12} - \phi_{13}) = -\frac{1}{\sqrt{2}}, \cos(\nu_1 - \phi_{13}) = 0$. Hence, $\phi_{11} = \nu_1$, and $\phi_{12} = \nu_1 - \pi/4, \phi_{13} = \nu_1 + \pi/2$ or $\phi_{12} = \nu_1 + \pi/4, \phi_{13} = \nu_1 - \pi/2$.

Then, for $\beta_2, \cos(\nu_2 - \phi_{22}) = -\frac{1}{\sqrt{2}}, \cos(\phi_{22} - \phi_{23}) = 0, \cos(\nu_2 - \phi_{23}) = \frac{1}{\sqrt{2}}$. Hence, $\phi_{21} = \nu_2$, and $\phi_{22} = \nu_2 + 3\pi/4, \phi_{23} = \nu_2 + \pi/4$ or $\phi_{22} = \nu_2 - 3\pi/4, \phi_{23} = \nu_2 - \pi/4$. Finally, for $\beta_3, \cos(\nu_3 - \phi_{32}) = 0, \cos(\phi_{32} - \phi_{33}) = -\frac{1}{\sqrt{2}}, \cos(\nu_3 - \phi_{33}) = \frac{1}{\sqrt{2}}$. Hence, $\phi_{31} = \nu_3$, and $\phi_{32} = \nu_3 + \pi/2, \phi_{33} = \nu_3 + 3\pi/4$ or $\phi_{32} = \nu_3 - \pi/2, \phi_{33} = \nu_3 - \pi/4$. We have from equation (7), (8) and (10) that

$$\psi_{\beta_1} = \sum_i \sqrt{P_{\alpha_1} \beta_{\alpha_1}} e^{i(\xi_{\alpha_1} + \theta_{\beta_{\alpha_1}})}$$

where $\xi_{\alpha_1} = 0$ and $\sqrt{P_{\alpha_1} \beta_{\alpha_1}} = \frac{1}{3}$. Thus

$$\psi_{\beta_1} = \frac{1}{3} (e^{i\nu_1} + e^{i\nu_1 + \pi/4} + e^{i\nu_1 \pm \pi/2}) = \frac{e^{i\nu_1}}{3}((1 + e^{i \pi/4} + e^{i \pm \pi/2}) = \frac{e^{i\nu_1}}{3}((1 + \sqrt{2}/2) \pm i(1 - \sqrt{2}/2));$$

$$\psi_{\beta_2} = \frac{e^{i\nu_2}}{3}(1 + i \sqrt{2})\text{ and, finally, } \psi_{\beta_3} = \frac{e^{i\nu_3}}{3}((1 - i \sqrt{2}) \pm i(1 + \sqrt{2}/2)).$$

We remark that $|\psi_{\beta_j}|^2 = 1/3, j = 1, 2, 3$. 

14
QLRA produces following possible realizations of the “wave function”:
\[
\psi = \frac{1}{3} \left[ ((1 + \frac{1}{\sqrt{2}}) \pm i(1 - \frac{1}{\sqrt{2}})) e_{\beta_1}^b + (1 \pm i\sqrt{2}) e_{\beta_2}^b + ((1 - \frac{1}{\sqrt{2}}) \pm i(1 + \frac{1}{\sqrt{2}})) e_{\beta_3}^b \right], \quad (75)
\]
where \( e_{\beta}^b \) are the orthonormal basis
\[
e_{\beta_1}^b = \begin{pmatrix} e^{i\nu_1} \\ 0 \\ 0 \end{pmatrix}, \quad e_{\beta_2}^b = \begin{pmatrix} 0 \\ e^{i\nu_2} \\ 0 \end{pmatrix}, \quad e_{\beta_3}^b = \begin{pmatrix} 0 \\ 0 \\ e^{i\nu_3} \end{pmatrix}. \quad (76)
\]
For the minus-case (i.e. \( \mu = -\frac{1}{\sqrt{2}} \)) QLRA produces following “wave function”:
\[
\psi = \frac{1}{3} \left[ ((1 - \frac{1}{\sqrt{2}}) \pm i(1 + \frac{1}{\sqrt{2}})) e_{\beta_1}^b + (1 \pm i\sqrt{2}) e_{\beta_2}^b + ((1 + \frac{1}{\sqrt{2}}) \pm i(1 - \frac{1}{\sqrt{2}})) e_{\beta_3}^b \right]. \quad (77)
\]

Summary

First we have shown that Sorkin’s equality can be derived for conditional probabilistic form by QRLA. The reconstruction of quantum states is for the case of trichotomous observables a complex task, but still we have managed to show that QLRA reconstruct at least a selection of quantum states for the case of trichotomous observables. For this task we have used mutually unbiased bases, thus we have found and calculated the probabilistic data. We have estimated the interference and shown that there exist positive probabilities (probabilistic data) for this choice of bases. We have also given some more selections of bases and analyses them numerical.

Acknowledgments

One of the authors, Irina Basieva, is supported by Swedish Institute, post-doc fellowship. We would like to thank professor Andrei Khrennikov for fruitful discussions about quantum formalism and probability theory.

References


Peter Nyman
Peter Nyman

On hyperbolic interferences in the quantum–like representation algorithm for the case of triple–valued observables

Received: date / Accepted: date

Abstract The quantum–like representation algorithm (QLRA) was introduced by A. Khrennikov [1–5] to solve the “inverse Born’s rule problem”, i.e. to construct a representation of probabilistic data– measured in any context of science– and represent this data by a complex or more general probability amplitude which matches a generalization of Born’s rule. The outcome from QLRA will introduce the formula of total probability with an additional term of trigonometric, hyperbolic or hyper-trigonometric interference and this is in fact a generalization of the familiar formula of interference of probabilities.

We study representation of statistical data (of any origin) by a probability amplitude in a complex algebra and a Clifford algebra (algebra of hyperbolic numbers). The statistical datas are collected from measurements of two trichotomous observables and the complexity of the problem increased eventually compared to the case of dichotomous observables. We see that only special statistical data (satisfying a number of nonlinear constraints) have a quantum–like representation. In this paper we will present a class of statistical data which satisfy these nonlinear constraints and have a quantum–like representation. This quantum–like representation induces trigonometric-, hyperbolic- and hyper–trigonometric interferences representation.

Keywords Hyperbolic interferences · Quantum–like representation algorithm · Clifford algebra · Born’s rule · Hyperbolic Hilbert space

F. Author
International Center for Mathematical Modeling in Physics and Cognitive Sciences, Linnaeus University, S-35195, Sweden
E-mail: peter.nyman@lnu.se

1 A Clifford algebra is introduced for this more general representation
1 Introduction

The inter-relation between classical and quantum probabilistic data was discussed in numerous papers (from various points of view), see\(^2\), e.g., [6–15]. We are interested in the representation of probabilistic data and of any origin\(^3\) by a generalization of complex probability amplitudes (wave function). This problem was discussed in detail in [27]. Here we describe a class of probabilistic data which permits the quantum-like (QL) representation for the case of two trichotomous observable. Recently the interest to trichotomous observable increased in connection with experiments of Greger Weih’s groups to check validity of Born’s rule for triple slit experiment [17]. This test was proposed by R.D. Sorkin [18,19] and this is an important test of foundation of quantum mechanics. Experimental studies [17] are characterized by essential increasing of complexity compared to the well known two slit experiment. We met the same increasing of complexity in our general theoretical study.

A general QL-representation algorithm (QLRA) was presented in [27]. This algorithm is based on the formula of total probability with interference term – a disturbance of the standard formula of total probability. Starting with experimental probabilistic data, QLRA produces a complex probability amplitude such that probability can be reconstructed by using Born’s rule.

Although the formal scheme of QLRA works for multi-valued observables of an arbitrary dimension, the description of the class of probabilistic data which can be transferred into QL-amplitudes (the domain of application of QLRA) depends very much on the dimension. In [29] the simplest case of data generated by dichotomous observables was studied. Examples of probabilistic data that generate QL-amplitudes are studied in [16]. The complexity of the problem increases incredibly compared to the two dimensional case, but here we show a class of probabilistic data that generate QL-amplitudes.

It appears naturally to represent quantum physics by hyperbolic numbers (also known as perplex, unipodal, duplex or split-complex algebra) and have application to physics [21]. Hucks shows that a four-component Dirac spinor is equivalent to a two-component hyperbolic complex spinor and that Lorentz group is homomorphic to the hyperbolic unitary group [22]. Ulrych analysis the symmetry of the hyperbolic Hilbert space and representation of Poincaré mass operator in the hyperbolics algebra [23].

The scheme of presentation is the following. We start with observables given by quantum mechanics (QM) and derive constraints on phases which are necessary and sufficient for the QL-representation. Then we use these constraints to produce complex amplitudes and more general (so called hyperbolic) amplitudes from data (of any origin); some examples, including numerical, are given. In this paper we stress the role of hyperbolic amplitudes, i.e., amplitudes valued in a special Clifford algebra, so called hyperbolic algebra, see e.g. [20].

---

\(^2\) The list of references is far from complete, see Khrennikov’s monographs [15,27] for a detailed list of references.

\(^3\) Thus it need not be produced by quantum measurements; it can be collected in e.g. psychology, see [28].
2 Trichotomous incompatible quantum observables

2.1 Probabilities

Let \( \hat{a} \) and \( \hat{b} \) be two self-adjoint operators in three dimensional complex Hilbert space representing two trichotomous incompatible observables \( a \) and \( b \). They take values \( a = \alpha_i, i = 1, 2, 3 \) and \( b = \beta_j, j = 1, 2, 3 \) – spectra of operators. We assume that the operators have non-degenerated spectra, i.e., \( \alpha_i \neq \alpha_j, \beta_k \neq \beta_j, i \neq j \). Consider corresponding eigenvectors:

\[
\hat{a} e_{\alpha_i}^a = \alpha_i e_{\alpha_i}^a, \quad \hat{b} e_{\beta_j}^b = \beta_j e_{\beta_j}^b.
\]

Denote by \( P^a_{\alpha_i} \) and \( P^b_{\beta_j} \) one dimensional projection operators and by \( P^a_{\alpha_i} \) and \( P^b_{\beta_j} \) the observables represented by these projections. Consider also projections

\[
P^a_{\alpha_i} = P^a_{\alpha_1} + P^a_{\alpha_2} + P^a_{\alpha_3}, \quad P^b_{\beta_j} = P^b_{\beta_1} + P^b_{\beta_2} + P^b_{\beta_3}.
\]

Here the observable \( P^a_{\alpha_1} \) is defined in the same way. We have the following relation between events corresponding to measurements

\[
[P^a_{\alpha_i} = 0] = [P^a_{\alpha_i} = 1] \lor [P^a_{\alpha_j} = 1], \quad [P^a_{\alpha_i} = 0] = [P^a_{\alpha_i} = 1] \lor [P^a_{\alpha_j} = 1], \quad (2)
\]

Here the probabilities given by the QM-formalism are

\[
p^a_\alpha \equiv P_\psi(b = \alpha) = ||P^a_{\alpha} \psi||^2 = ||(\psi e^a_\alpha)||^2, \quad (3)
\]

\[
p^b_\beta \equiv P_\psi(a = \beta) = ||P^b_{\beta} \psi||^2 = ||(\psi e^b_\beta)||^2,
\]

where \( \psi \) is a wave function. We also have the conditional (transition) probabilities given by the QM-formalism as

\[
p^{b\alpha}_{\beta \alpha} \equiv P_\psi(b = \beta | a = \alpha) = ||P^b_{\beta} P^a_{\alpha} \psi||^2 / ||P^a_{\alpha} \psi||^2 = ||(e^a_\alpha e^b_\beta)||^2. \quad (4)
\]

We remark that non-degeneration of the spectra implies that they do not depend on \( \psi \). Moreover, the matrix of transition probabilities is \textit{doubly stochastic}. We will require the following conditions (compare this with classical probability theory) for these conditional probabilities;

\[
\sum_{i=1}^{3} p^{b\alpha}_{\beta \alpha_i} = 1, \quad (5)
\]

for all \( j = 1, 2, 3 \). Here the \( \psi \)-dependent conditional probabilities are

\[
p^{b\alpha}_{\beta \alpha_j} \equiv P_\psi(b = \beta_j | a = \alpha_j) = 1 \lor P^a_{\alpha_j} = 1 = P_\psi(b = \beta_j | a = \alpha_j) = 0
\]

\[
= \frac{||P^b_{\beta_j} (P^a_{\alpha_j} + P^a_{\alpha_j}) \psi||^2}{|| (P^a_{\alpha_j} + P^a_{\alpha_j}) \psi||^2}, \quad (6)
\]
where \( i,j,k \in \{1, 2, 3\}, j, k \neq i \). We have
\[
\frac{|| P_β (\hat{P}_α^j a_j + \hat{P}_α^k a_k) \psi ||^2}{|| (\hat{P}_α^j a_j + \hat{P}_α^k a_k) \psi ||^2} = \frac{|| (\hat{P}_β^b \hat{P}_α^j a_j + \hat{P}_β^b \hat{P}_α^k a_k) \psi ||^2}{|| (\hat{P}_α^j \hat{P}_α^k a_j + \hat{P}_α^k \hat{P}_α^j a_k) \psi ||^2} \neq 0 \text{ and } x+y \frac{x+y}{x-y} > 0, \text{ since } z \in G^*.
\]

We also define a hyperbolic exponential function
\[
e_{jθ} = \cosh θ + j \sin θ, \quad θ \in \mathbb{R}.
\]

\section{3 Clifford algebra (hyperbolic algebra)}

As mention before we will consider the complex Hilbert space but also the hyperbolic Hilbert space. Therefore let us define a Clifford algebra called hyperbolic algebra (see \cite{27}) with the purpose to define the hyperbolic Hilbert space. The formalism for this hyperbolic algebra is similar to conventional complex numbers. This algebra contains expressions as unit circle, Euler’s formula and conjugate. Thus, let an element \( z \) belong to the hyperbolic algebra \( G \) if and only if it have following form;
\[
z = x + jy, \quad x, y \in \mathbb{R},
\]
where \( j^2 = 1 \), \( z_1 + z_2 = z_1 + z_2 + j(y_1 + y_2) \) and \( z_1 z_2 = z_1 z_2 + j(y_1 z_2 + y_2 z_1) \). This algebra is a commutative two-dimensional algebra with two orthonormal basis \( e_0 = 1 \) and \( e_1 = 1 \). The hyperbolic conjugate is defined as \( \bar{z} = x - jy \) where obviously \( \bar{\bar{z}} = z \). Moreover, the square of absolute value is defined by
\[
|z|^2 = z\bar{z} = x^2 - y^2
\]
and therefore will \( |z|^2 \in G \), in fact \( |z|^2 \in \mathbb{R} \). But \( |z| \) will not be well defined for \( z \) such that \( |z|^2 \leq 0 \). Therefore set
\[
G_+ = \{ z \in G : |z|^2 \geq 0 \}.
\]
and
\[
G^*_+ = \{ z \in G : |z|^2 > 0 \}.
\]
Moreover define the argument \( \arg z \) for \( z \in G^*_+ \) as
\[
\arg z = \arctanh \frac{y}{x} = \frac{1}{2} \ln \frac{x+y}{x-y}.
\]
Notice that \( x \neq 0, x - y \neq 0 \) and \( \frac{x+y}{x-y} > 0 \), since \( z \in G^*_+ \). We also define a hyperbolic exponential function
\[
e^{jθ} = \cosh θ + j \sin θ, \quad θ \in \mathbb{R}.
\]
Since $\cosh \theta \geq 0$, elements $z \in G_+^*$ with $x < 0$ cannot be represented by $|z|e^{i\theta}$. Therefore, in order to represent all elements $z \in G_+^*$ put

$$z = \varepsilon |z|e^{i\theta},$$

where $\varepsilon = x/|x|$ and $\arg z = \theta$. Moreover, this is a multiplicative semigroup. Let $z_1, z_2 \in G_+^*$, so $|z_1|^2, |z_2|^2 > 0$ then we see that $z_1 \cdot z_2 \in G_+^*$ by

$$|z_1z_2|^2 = |e_1|z_1|e^{i\theta_1}\cdot e_2|z_2|e^{i\theta_2}|^2 = |z_1|^2|z_2|^2 > 0.$$

But, when we add two elements $z_1, z_2 \in G_+^*$ it follows that the existence of elements so that $z_1 + z_2 \notin G_+^*$. Let us analyze for which of the elements $z_1, z_2 \in G_+^*$, $z_1 + z_2 \notin G_+^*$, i.e. $|z_1 + z_2|^2 < 0$. It follows that

$$|z_1 + z_2|^2 = |e_1|z_1|e^{i\theta_1} + e_2|z_2|e^{i\theta_2}|^2 = |z_1|^2 + |z_2|^2 + 2e_1e_2|z_1||z_2|\cosh(\theta_1 - \theta_2) \quad (8)$$

From (8) and $\cosh(\theta_1 - \theta_2) > 0$ it follows that $|z_1 + z_2|^2 > 0$ if $e_1e_2 = 1$. Here we consider elements $z_1, z_2 \in G_+^*$ such that $|z_1|^2 + |z_2|^2 + 2e_1e_2|z_1||z_2|\cosh(\theta_1 - \theta_2) > 0$. Let $e_1e_2 = -1$ then $|z_1 + z_2|^2 > 0$ if and only if

$$\arccosh\left(\frac{|z_1|^2 + |z_2|^2}{2|z_1||z_2|}\right) > |\theta_1 - \theta_2|.$$

3.1 Hyperbolic Hilbert space

A hyperbolic Hilbert space $H$ is a $G$-linear inner product space. Let $x, y, z \in H$ and $a, b \in G$, then consider the inner product as a map from $H \times H$ to $G$ having the following properties:

1. Conjugate symmetry: $\langle x, y \rangle$ is the conjugate to $\langle y, x \rangle$

$$\langle x, y \rangle = \overline{\langle y, x \rangle}$$

2. Linearity with respect to the first argument:

$$\langle ax + bz, y \rangle = a \langle x, y \rangle + b \langle z, y \rangle$$

3. Non-degenerate:

$$\langle x, y \rangle = 0$$

for all $y \in H$ if and only if $x = 0$

In general, the norm $\| \psi \| = \sqrt{\langle \psi, \overline{\psi} \rangle}$ is not well defined. But we only need the square of the norm $\| \psi \|^2 = \langle \psi, \psi \rangle$. 
3.2 Probability amplitudes

Set \( \psi_\beta = \langle \psi | e^b_\beta \rangle \), and consider the complex Hilbert space and the hyperbolic Hilbert space. Then by Born’s rule

\[
p^b_\beta = |\psi_\beta|^2. \tag{9}
\]

We have

\[
\psi = \sum_\beta \psi_\beta e^b_\beta. \tag{10}
\]

Thus these amplitudes give a possibility to reconstruct the state. We remark that \( \psi = \sum_\alpha \hat{P}_a^\alpha \psi \), hence

\[
\psi_\beta = \sum_\alpha \langle \hat{P}_a^\alpha \psi | e^b_\beta \rangle. \tag{11}
\]

Each amplitude \( \psi_\beta \) can be represented as the sum of subamplitudes

\[
\psi_\beta = \sum_\alpha \psi_{\beta\alpha} \tag{12}
\]

given by

\[
\psi_{\beta\alpha} = \langle \hat{P}_a^\alpha \psi | e^b_\beta \rangle = \langle \psi | e^a_\alpha \rangle \langle e^a_\alpha | e^b_\beta \rangle. \tag{13}
\]

Hence, one can reconstruct the state \( \psi \) on the basis of amplitudes \( \psi_{\beta\alpha} \). We remark that \( |\psi_{\beta\alpha}|^2 = |\langle \psi | e^a_\alpha \rangle \langle e^a_\alpha | e^b_\beta \rangle|^2 = p^a_\alpha p^b_\beta \delta_{\alpha\beta} \).

In this notations

\[
p^b_\beta_{\alpha\beta} = |\psi_{\beta\alpha}|^2 / (p^a_\alpha + p^a_\beta). \tag{14}
\]

Here \( |\psi_{\beta\alpha}| = \sqrt{p^a_\alpha p^b_\beta} \) and therefore

\[
\psi_{\beta\alpha} = \sqrt{p^a_\alpha p^b_\beta} \lambda_{\psi_{\beta\alpha}}, \tag{15}
\]

where \( |\lambda_{\psi_{\beta\alpha}}| = 1 \). Moreover, put

\[
\langle \psi | e^a_\alpha \rangle = \sqrt{p^a_\alpha} \lambda_{\psi_{\alpha}}, \quad \langle e^a_\alpha | e^b_\beta \rangle = \sqrt{p^b_\beta} \lambda_{\psi_{\beta}}. \tag{16}
\]

where \( |\lambda_{\psi_{\alpha}}| = 1 \) and \( |\lambda_{\psi_{\beta}}| = 1 \). Hence, it follows from (13) and (15) that

\[
\lambda_{\psi_{\beta\alpha}} = \lambda_{\psi_{\alpha}} \lambda_{\psi_{\beta}}. \tag{17}
\]
We have a system of equations for phases $\psi_{\beta l \alpha i}$ for $i, j, k, l \in \{1, 2, 3\}$,

$$|\psi_{\beta l \alpha i} + \psi_{\beta l \alpha j}|^2 = |\langle \psi | e^a_{\alpha i} e^b_{\alpha j} \rangle|^2 \quad (18)$$

Thus, the coefficients of interference $\lambda_{i,j}$ can be written by (14) and (18) as

$$\lambda_{i,j} \equiv \frac{2\lambda_{i,j}}{P_{\alpha i}P_{\beta i}P_{\alpha j}P_{\beta j}}$$

3.3 Interference Classification

Note that the coefficients of interference in (20) will take values in $\mathbb{R}$. We divide this into following two cases of interference depending on the absolute value of $\lambda_{i,j}$.

1. Let $|\lambda_{i,j}| \leq 1$ and in this case put $\lambda_{\phi_{\beta l \alpha i} \phi_{\beta l \alpha j}} = e^{\phi_{\beta l \alpha i}}$, then it straightforward from (17) that $e^{\phi_{\beta l \alpha i}} = e^{(\xi_{\alpha i} + \theta_{\alpha i})}$. Thus, by (19) we see that

$$\lambda_{i,j} = \cos(\phi_{\beta l \alpha i} - \phi_{\beta l \alpha j}).$$

We refer to this interference as trigonometric interference.

2. Let $|\lambda_{i,j}| > 1$ and put $\lambda_{\phi_{\beta l \alpha i} \phi_{\beta l \alpha j}} = e_{i,j} e^{\phi_{\beta l \alpha i}}$ where $j^2 = 1$ and $e_{i,j} = \lambda_{\phi_{\beta l \alpha i}} / |\lambda_{\phi_{\beta l \alpha i}}|$. Here the symbol $j$ is a generator of the Clifford algebra $G_4^+$, have the form $z = x + jy$ and the “hyperbolic conjugate” $z = x - jy$. It is apparent that $z \in G_4^+$. We introduce the hyperbolic exponential function

$$e^{i\theta} = \cosh \theta + j \sinh \theta, \quad \theta \in \mathbb{R}. \quad (21)$$

4 Please note that this is not the usual complex conjugate! This is the analogous conjugate associated to the Clifford algebra. We are aware of the fact that this might be confusing, but would still like to call this conjugate because of its similarity to the complex conjugate.
We also use the identities
\[
\cosh \theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \quad \text{and} \quad \sinh \theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}.
\]
(22)

By (19) it follows that
\[
\lambda_{ij} = E_{ij} \cosh(\varphi_{\beta,\alpha} - \varphi_{\beta,\alpha}^*),
\]
where \(E_{ij} = E_iE_j\). We will call this hyperbolic interference.

3.4 Formula of total probability with interference term

By using the definition of the amplitude \(\psi_{\beta,\alpha} = \langle \psi | e^{ia} \rangle \langle e^{ia} | \psi \rangle\) we obtain
\[
P_{\beta_i}^h = |\psi_{\beta,\alpha} + |\psi_{\beta,\alpha} + |\psi_{\beta,\alpha}|^2\]

(23)

\[
= |\langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle|^2 + |\langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle|^2 + |\langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle|^2 +
\]

\[
+ \langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle |\psi_{\beta,\alpha} |^2 |\psi_{\beta,\alpha} |(\psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle) + \langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle |\psi_{\beta,\alpha} |^2 |\psi_{\beta,\alpha} |
\]

0

+ \langle \psi_{\beta,\alpha} | e^{ia} \rangle \langle e^{ia} | \psi_{\beta,\alpha} \rangle |\psi_{\beta,\alpha} |^2 |\psi_{\beta,\alpha} |

Finally, we obtain
\[
P_{\beta_i}^h = p_{\alpha_i}^h p_{\beta,\alpha_i}^h + p_{\alpha_i}^h p_{\beta,\alpha_i}^h + p_{\alpha_i}^h p_{\beta,\alpha_i}^h + 2\lambda_{ij} \sqrt{p_{\alpha_i}^h p_{\beta,\alpha_i}^h p_{\beta,\alpha_i}^h p_{\beta,\alpha_i}^h}
\]

(24)

Here, if \(|\lambda_{ij}| \leq 1\) for all \(l, l, j = 1, 2, 3, i \neq j\) then we call this the case of trigonometric interference and the case where \(|\lambda_{ij}| > 1\) for all \(l, l, j = 1, 2, 3, i \neq j\) is called the case of hyperbolic interference. All other cases are combinations of these two cases of interference and is called the hyper-trigonometric interference (i.e., for some \(l, l, j = 1, 2, 3, i \neq j, |\lambda_{ij}| < 1\) and for the rest \(|\lambda_{ij}| > 1\)). Equation (24) is nothing else than the formula of total probability with the interference terms. It can be considered [30] as a perturbation of the classical formula of total probability
\[
P_{\beta_i}^h = p_{\alpha_i}^h p_{\beta,\alpha_i}^h + p_{\alpha_i}^h p_{\beta,\alpha_i}^h + p_{\alpha_i}^h p_{\beta,\alpha_i}^h.
\]

(25)

If all coefficients of interferences \(\lambda_{ij} = 0\), then (24) coincides with (25).
3.5 Unitarity of transition operator

We now remark that bases consisting of $\hat{a}$- and $\hat{b}$-eigenvectors are orthogonal; hence the operator of transition from one basis to another is unitary. We can always select the $b$-basis in the canonical way

$$e^b_{\beta_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad e^b_{\beta_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad e^b_{\beta_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (26)$$

In this system of coordinates the $a$-basis has the form

$$e^a_{\alpha_1} = \begin{pmatrix} \sqrt{p_{\beta_1\alpha_1}} \lambda_{\phi_{\beta_1\alpha_1}} \\ \sqrt{p_{\beta_2\alpha_1}} \lambda_{\phi_{\beta_2\alpha_1}} \\ \sqrt{p_{\beta_3\alpha_1}} \lambda_{\phi_{\beta_3\alpha_1}} \end{pmatrix}, \quad e^a_{\alpha_2} = \begin{pmatrix} \sqrt{p_{\beta_1\alpha_2}} \lambda_{\phi_{\beta_1\alpha_2}} \\ \sqrt{p_{\beta_2\alpha_2}} \lambda_{\phi_{\beta_2\alpha_2}} \\ \sqrt{p_{\beta_3\alpha_2}} \lambda_{\phi_{\beta_3\alpha_2}} \end{pmatrix}, \quad e^a_{\alpha_3} = \begin{pmatrix} \sqrt{p_{\beta_1\alpha_3}} \lambda_{\phi_{\beta_1\alpha_3}} \\ \sqrt{p_{\beta_2\alpha_3}} \lambda_{\phi_{\beta_2\alpha_3}} \\ \sqrt{p_{\beta_3\alpha_3}} \lambda_{\phi_{\beta_3\alpha_3}} \end{pmatrix}. \quad (27)$$

The matrix

$$U = \begin{pmatrix} \sqrt{p_{\beta_1\alpha_1}} \lambda_{\phi_{\beta_1\alpha_1}} & \sqrt{p_{\beta_1\alpha_2}} \lambda_{\phi_{\beta_1\alpha_2}} & \sqrt{p_{\beta_1\alpha_3}} \lambda_{\phi_{\beta_1\alpha_3}} \\ \sqrt{p_{\beta_2\alpha_1}} \lambda_{\phi_{\beta_2\alpha_1}} & \sqrt{p_{\beta_2\alpha_2}} \lambda_{\phi_{\beta_2\alpha_2}} & \sqrt{p_{\beta_2\alpha_3}} \lambda_{\phi_{\beta_2\alpha_3}} \\ \sqrt{p_{\beta_3\alpha_1}} \lambda_{\phi_{\beta_3\alpha_1}} & \sqrt{p_{\beta_3\alpha_2}} \lambda_{\phi_{\beta_3\alpha_2}} & \sqrt{p_{\beta_3\alpha_3}} \lambda_{\phi_{\beta_3\alpha_3}} \end{pmatrix}$$

is unitary. Hence, we have the system of equations

$$\sum_m p_{\beta_\alpha} p_{\beta_\alpha} \lambda_{\phi_{\beta\alpha}} \overline{\lambda_{\phi_{\beta\alpha}}} = 0 \quad (28)$$

and

$$\sum_m p_{\beta_\alpha} |\lambda_{\phi_{\beta\alpha}}|^2 = 1, \quad (29)$$

from the condition that $U$ is unitary.

The second equations (29) will always hold by (5) and $|\lambda_{\phi_{\beta\alpha}}| = 1$, see (15).

The first of this equations (28) can be rewritten by (17) as

$$\lambda_{\phi_{\beta_1\alpha_1}} \overline{\lambda_{\phi_{\beta_{\alpha_1}}} \overline{\lambda_{\phi_{\beta_{\alpha_1}}}}} = 0, \quad (30)$$

or

$$\sum_m \sqrt{p_{\beta_\alpha} p_{\beta_\alpha}} \lambda_{\phi_{\beta_\alpha}} \overline{\lambda_{\phi_{\beta_\alpha}}} = 0, \quad (31)$$

where $\lambda_{\phi_{\beta_1\alpha_1}}, \lambda_{\phi_{\beta_1\alpha_1}} \neq 0$.

Thus (31) imply unitarity of $U$. 
4 Selection of orthonormal bases for the unitarity of transition operator

We now show by example that there exist unitary $U$ satisfying QLRA. Let us choose to work with the case of hyperbolic interference. The same calculations with similar basis can be done with trigonometric interference. When repeating these calculations for trigonometric interference put $\lambda_{\phi|^\alpha} = e^{i\theta_{\psi^\alpha}}$. Choose orthonormal $\psi^\alpha$-bases by setting $\lambda_{\phi|^\alpha} = e^{i\theta_{\psi^\alpha}}, p_{\alpha\beta} = a_{\alpha\beta}^2/(1 + a_{\alpha\beta}^2 + a_{\alpha\beta}^2), \; \epsilon_{ii} = 1, \; \alpha_i = 1, \; \theta_{\psi^{\alpha_i}} = u, \; \theta_{\psi^{\alpha_i}} = s, \; \theta_{\psi^{\alpha_i}} = t, \; i = 1, 2, 3$ in (27):

$$
e_{\alpha}^i = \frac{1}{\sqrt{1 + a_{\alpha\beta}^2 + a_{\alpha\beta}^2}} \left( e^{i\theta_{\psi^\alpha}} e_{\lambda u}^{j\mu} \right).
$$

Since the bases are orthonormal we have that

$$
\langle e_{\alpha}^i | e_{\alpha}^j \rangle = \begin{cases} 
1 + a_{\alpha\beta}^2 e_{\lambda 2} e_{\lambda 3} + a_{\alpha\beta}^3 e_{\lambda 3} & \text{if } k \neq i, \\
1 & \text{if } i = k.
\end{cases}
$$

where $i, k = 1, 2, 3$. The case $k \neq i$ give us a system of equations $1 + a_{\alpha\beta}^2 e_{\lambda 2} e_{\lambda 3} + a_{\alpha\beta}^3 e_{\lambda 3} = 0$, $i, k = 1, 2, 3$, with solution

$$
a_{31} = \frac{-a_{31}^2 e_{\lambda 2}^2 - a_{32} a_{33} e_{\lambda 2} e_{\lambda 3} - 1}{e_{\lambda 1} \left( a_{31}^2 e_{\lambda 2} e_{\lambda 2} + a_{32} a_{33} e_{\lambda 2} e_{\lambda 1} + a_{33} e_{\lambda 3} \right)}, \\
a_{31} = \frac{-a_{32} e_{\lambda 2} e_{\lambda 2} - a_{33} e_{\lambda 3}}{e_{\lambda 1} \left( a_{31}^2 e_{\lambda 2} e_{\lambda 2} + a_{32} a_{33} e_{\lambda 2} e_{\lambda 1} + a_{33} e_{\lambda 3} \right)}, \\
a_{32} = \frac{-a_{32} a_{33} e_{\lambda 2} e_{\lambda 3} - 1}{a_{32} e_{\lambda 2} e_{\lambda 2}}.
$$

Then let

$$
\psi = \frac{1}{\sqrt{v_1 e^{i\theta_{\psi^\alpha}} + v_2 e^{i\theta_{\psi^\alpha}} + v_3 e^{i\theta_{\psi^\alpha}}}}.
$$

Note that $|\psi|^2 = 1$. Next we calculate the probabilities $p_{\alpha} = |\langle e_{\alpha}^i | \psi \rangle|^2$ for $i = 1, 2, 3$, in consideration of solutions (32). In order to reduce the size of the expressions we introduce the following rewriting.

$$
d_1 = (1 + a_{32}^2 + a_{32} a_{31} e_{\lambda 2} e_{\lambda 3}), \quad d_2 = (a_{32}^2 (a_{32}^2 + a_{31}^3) e_{\lambda 2} + a_{33} e_{\lambda 3}), \\
d_3 = (1 + a_{32} a_{33} e_{\lambda 2} e_{\lambda 3}), \quad d_4 = a_{32} e_{\lambda 2} e_{\lambda 2} - a_{33} e_{\lambda 3}, \\
d_5 = (1 + a_{32} a_{33} e_{\lambda 2} e_{\lambda 3}), \quad d_6 = a_{32} v_3 e_{\lambda 2}, \\
d_7 = a_{32} v_3 e_{\lambda 2}, \quad d_8 = a_{33} v_3 e_{\lambda 3}, \\
d_9 = a_{32}^2 v_3^2 e_{\lambda 2}, \quad d_{10} = (a_{32} e_{\lambda 2} + a_{32} a_{33} e_{\lambda 2} e_{\lambda 3})^2, \\
d_{11} = 1 + a_{32}^2 + a_{33}^2, \quad d_{12} = a_{32}^2 (1 + a_{32}^2) + a_{33}^2, \\
d_{13} = v_1^2 + v_2^2 + v_3^2, \quad \gamma_{12} = \gamma - \gamma + s - u, \\
\gamma_{13} = \gamma - \gamma + t - u.
$$
We then get the following probabilities,

\[ p_{\alpha_1} = (d_1^2 + 2\cosh(\gamma_{12}) d_4 v_2 + \cosh(\gamma_{13}) d_4 v_3) v_1 + d_3 v_2^2 \\
+ d_1 v_3 (2\cosh(\gamma_{12} - \gamma_{13}) d_4 v_2 + d_1 v_3) (d_{11} d_{12} d_{13})^{-1}, \]

\[ p_{\alpha_2} = (v_1^2 + 2\cosh(\gamma_{13}) d_4 v_1 + d_3 v_2^2 + d_5 v_2^3) d_{23}^{-2} \\
- 2 d_5 (\cosh(\gamma_{12} - \gamma_{13}) d_6 + \cosh(\gamma_{12}) v_1) v_2 e_{23} a_{23} + d_5 v_2^3 (d_{12} d_{13})^{-1}, \]

\[ p_{\alpha_3} = (v_1^2 + 2\cosh(\gamma_{13}) d_4 v_1 + d_3 v_2^2 + d_5 v_2^3) (d_{11} d_{12} d_{13})^{-1}, \]

\[ 2 d_7 (\cosh(\gamma_{12} - \gamma_{13}) d_8 + \cosh(\gamma_{12}) v_1) (d_{11} d_{12} d_{13})^{-1} \]

The calculations of the conditional probabilities \( p_{\beta a_i}^{\beta a_j} \) are found in the appendix. Moreover

\[ p_{\beta a_i}^{\beta a_j} = \frac{d_i^2}{d_{11} d_{12}}, p_{\beta a_2}^{\beta a_3} = \frac{1}{d_{32}^2 + d_{32}^2 + 1}, p_{\beta a_3}^{\beta a_3} = \frac{1}{d_{11}} \]

\[ p_{\beta a_i}^{\beta a_j} = \frac{d_3}{d_{11} d_{12}}, p_{\beta a_2}^{\beta a_3} = \frac{d_5^2}{d_{11} d_{12}}, p_{\beta a_3}^{\beta a_3} = \frac{d_3^2}{d_{11}} \]

\[ p_{\beta a_i}^{\beta a_j} = \frac{d_1^2}{d_{11} d_{12}}, p_{\beta a_2}^{\beta a_3} = \frac{d_3^2}{d_{32}^2 + d_{32}^2 + 1}, p_{\beta a_3}^{\beta a_3} = \frac{d_3^2}{d_{11}} \]

Please note that we require that these probabilities satisfy 0 < \( p_{\beta a_i}^{\beta a_j} < 1, 0 < p_{\alpha i} < 1 \). Since these probabilities also satisfy that \( \sum_i p_{\alpha_i} = 1, \sum_j p_{\beta a_i} = 1 \), it is sufficient to show that \( \sum_i p_{\alpha_i} > 0 \) and \( \sum_j p_{\beta a_i} > 0 \) for all \( k, j \). In order to show that there exist at least one such orthonormal basis in (4) and \( \psi \) in (33) that meets these requirements on the probabilities, we here give an example. Let \( \gamma_1 = 0, \gamma_2 = t, \gamma_3 = t, v_1 = -2, v_2 = 3, v_3 = -2 \), in the quantum state in (33)

\[ \psi = \frac{1}{\sqrt{17}} (-2 e_{\beta 1} + 3 e_{\beta 2} - 2 e_{\beta 3}). \]  

(35)

Let \( e_{23} a_{23} = 2, e_{32} a_{32} = 2, e_{33} a_{33} = 3, s = t, u = 0.3 \) in the basis in (4)

\[ e_{a 1}^a = \frac{1}{\sqrt{1 + (2)^2 + (11)^2}} \left( e_{0}^0 + \frac{2}{11} e_{1}^0 \right), e_{a 2}^a = \frac{1}{\sqrt{1 + (\frac{7}{2})^2 + (2)^2}} \left( e_{0.3}^j - \frac{2}{2} e_{0}^j \right), \]

\[ e_{a 3}^a = \frac{1}{\sqrt{1 + (2)^2 + (3)^2}} \left( e_{0}^2 + \frac{2}{3} e_{1}^2 \right). \]

Then

\[ p_{\alpha_1} = 0.045837, p_{\alpha_2} = 0.937356, p_{\alpha_3} = 0.016807, \]

\[ p_{\beta 1 a_1} = 0.206349, p_{\beta 1 a_3} = 0.887593, p_{\beta 1 a_2} = 0.075727, \]

\[ p_{\beta 2 a_1} = 0.650559, p_{\beta 2 a_3} = 0.111601, p_{\beta 2 a_2} = 0.580032, \]

\[ p_{\beta 3 a_1} = 0.143091, p_{\beta 3 a_3} = 0.000805, p_{\beta 3 a_2} = 0.344240. \]
5 Appendix:

Here we calculate the conditional probabilities

\[ P_{\beta_l \alpha_k j}^{\beta_l} = \left| \sum_{\alpha_j} P_{\beta_l \alpha_j}^{\beta_l} \right|^2 / \left( P_{\alpha_j}^{\alpha_j} + P_{\alpha_k}^{\alpha_k} \right). \]

We use (34) to reduce the size of the expressions

\[
\begin{align*}
|P_{\beta_l \alpha_k j}^{\beta_l}|^2 & = \frac{1}{1 + \left( (v_1^2 + v_3^2 + v_5^2 - 2\cosh(\gamma_{12} - \gamma_{13}) d_k v_2) \right)} \\
& \times \left[ (v_1^2 + v_3^2 + v_5^2 - 2\cosh(\gamma_{12}) d_k v_2) \right. \\
& - 2d_k (\cosh(\gamma_{12} - \gamma_{13}) d_k v_2 + a_{\alpha_j} (v_2 + v_5^2)) \left. \right]^{-1} \\

\end{align*}
\]
\[ \beta_{p_{12}}^{\mu_2} = \left( (a_1^2 + 1) a_2^3 + (a_1^2 + 1) a_2^2 + 2a_{13} \right)^2 v_1^2 \]

\[ - 2 \cosh (\gamma_{12} - \gamma_{13}) d_1 d_2 \left( (a_1^2 + 1) a_2^3 + (a_1^2 + 1) a_2^2 + 2a_{13} \right) v_1 v_2 \]

\[ + 2d_1 d_2 v_1 \left( \cosh (\gamma_{12}) \left( (a_1^2 + 1) a_2^3 + (a_1^2 + 1) d_1^2 + 2a_{13} \right) v_2 - \cosh (\gamma_{12}) d_1 d_2 v_1 \right) + d_1 \left( a_2^2 v_3^2 + a_2^2 v_1^2 \right) \]

\[ \left[ d_2 \left( \left( a_1^2 + 1 \right) a_2^3 + 2a_{23} \right) + \left( a_1^2 + 1 \right) a_2^2 + \left( a_2^3 + 1 \right) d_1^2 + 2a_{13} \right) v_1 \]

\[ - 2 \cosh (\gamma_{12} - \gamma_{13}) d_1 d_2 v_2 \]

\[ + v_1 \left( 2a_{23} \cosh \gamma_{23} a_2^3 + \left( a_1^2 + 1 \right) a_2^2 + a_2^3 \left( a_2^2 + \left( 2a_{12} + 1 \right) a_2^2 + a_{13} \right) \right) \]
References


Simulation of quantum algorithms on a symbolic computer.

Peter Nyman

Simulation of Quantum Algorithms on a symbolic computer

Peter Nyman

International Center for Mathematical Modeling in Physics, Engineering and Cognitive science
MSI, Växjö University, S-35195, Sweden

Abstract. This paper is a presentation of how to implement quantum algorithms (namely, Shor’s algorithm) on a classical computer by using the well-known Mathematica package. It will give us a lucid connection between mathematical formulation of quantum mechanics and computational methods.

Keywords: Shor’s factoring algorithm, prime factorization, quantum computing, Mathematica, quantum Fourier transform

PACS: 03.67.Lx, 02.70.-c, 07.05.Tp

INTRODUCTION

A couple of examples of various methods for the simulation of quantum algorithms were given in [1, 2]. In preprint [3] we introduced a computational language constructed on the basis of quantum mechanics. We have decided to implement the well-known Shor’s factoring algorithm as an example of a quantum algorithm. The aim is to construct a computational language in order to present a straightforward connection between Dirac’s mathematical formulation of quantum mechanics and the program code. Thus, the computational language will include quantum mechanical terminology such as quantum operators and quantum states.

THE SIMULATION FRAMEWORK

A quantum state in \( n \) dimensions can be represented by a linear combination of \( n \) numbers of basis vectors. In the two-dimensional case a quantum state \( |\phi\rangle \) is represented as a superposition of two basis vectors, say \( |0\rangle \) and \( |1\rangle \), (computation basis [4, 5]). In this case a quantum state \( |\phi\rangle \) is represented as

\[
|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,
\]

where \( \alpha \) and \( \beta \) are complex numbers having the sum of squares equal one.

We will introduce new symbols for states of this computational basis as follows: \( e[0] = |0\rangle \) and \( e[1] = |1\rangle \). This is the foundation for the structure of the program code. For more than one-qubit we will use the computational basis states \( e[x_1, \ldots, x_n] = |x_1 \ldots x_n\rangle \), where \( x_j \in \{0, 1\} \) or by using the more compact notation \( e[y] = |y\rangle \), where \( y = x_n2^n + \cdots + x_12^{n-1} \). We will, write the state \( \phi \) as \( e[\phi] = \alpha e[0] + \beta e[1] \), by analogy to
equation (1). The operator $A$ acting on the state $\phi$ is often written as $A|\phi\rangle$ in the quantum mechanical literature. To match these symbols, we will use the computational symbols $A[e|\phi]$ for this operation. The program code must be able to handle the linearity of tensor product. Let $e|v\rangle, e|w\rangle$ be vectors and $\alpha$ a scalar. We define the tensor product as

\[ \alpha(e|v\rangle \otimes e|w\rangle) = (\alpha e|v\rangle) \otimes (\alpha e|w\rangle) = e|v\rangle \otimes (\alpha e|w\rangle) \]

(2)

(3)

We add two commands to the program code that will implement this definition of the tensor product. The command $e[a_{\alpha}, \alpha, e|x\rangle, b_{\alpha}] := \alpha e[a, x, b]$ will transform $(\alpha e|v\rangle \otimes \alpha e|w\rangle)$ and $e|v\rangle \otimes (\alpha e|w\rangle)$ to $\xi e|a\rangle \otimes e|a\rangle \otimes \alpha e|x\rangle \otimes e|x\rangle$. The next command $e[a, \xi, e[x_1] + \beta e[y_1], b] := \xi e[a, x, b] + \xi \beta e[a, y, b]$ will transform $e[v_1] \otimes \xi(\alpha e|w_1\rangle + \beta e|w_2\rangle) \otimes e[v_1] \otimes e[w_1\rangle] + \xi \beta (e[v_1] \otimes e[w_1\rangle] + e[w_2\rangle] \otimes e[v_2\rangle])$. Let $U$ be an arbitrary one-qubit quantum gate. Then $U$ will transform an arbitrary state $e|\phi\rangle$ which is represented in the computational basis states as $e|\phi\rangle = a e|0\rangle + b e|1\rangle$ to the state $U|\phi\rangle \rightarrow (a e|0\rangle + c e|1\rangle) + b(c e|0\rangle + c e|1\rangle)$, where $a, b, c \in \mathbb{C}$. Now we add a Mathematica gate $U$ to the program code resulting in $U|e|0\rangle \rightarrow c_1 e|0\rangle + c_2 e|1\rangle$ and $U|e|1\rangle \rightarrow c_3 e|0\rangle + c_4 e|1\rangle$. For example, the Hadamard gate $H$ will be added in Mathematica as the command $H := \{ e[0] \rightarrow 1/\sqrt{2} (e[0] + e[1]), e[1] \rightarrow 1/\sqrt{2} (e[0] - e[1]) \}$. We will define a one-qubit gate $O_i$ as an operator which acts on the qubit in position $i$ and leaves the other qubits unchanged. The program code must be able to operate with a gate on arbitrary qubit. Consequently we will define an operator $O_i$ in the Mathematica code. The operator will be defined $O_i = I^{|i-1\rangle} \otimes \mathbb{C} \otimes I^{|i+n-1\rangle}$ as an operator which acts on $n$-qubits, where $I$ is the one-qubit unit operator and $U$ is an arbitrary one-qubit operator. Then operator $O_i$ is a function of $O_i|e|v\rangle \rightarrow e|v|$. Similarly, we will define $O_{ij}$ as an operator which operates as the two-qubits operator on the qubits in positions $i, j$ and leaves the other qubits unchanged. Now we have the tools to build quantum circuits. To achieve this, we will use a quantum Fourier transform circuit in Shor’s factoring algorithm.

AN INTRODUCTION TO SHOR’S FACTORING ALGORITHM

Prime factoring of an odd number $N$ can be accomplished using Shor’s algorithm [6]. If $N$ is an even integer then we divide it with 2 $n$-times until $2^{-n}N$ becomes an odd number. An even $N = 2n$ can easily be found in view of the fact that $2n \equiv 0 \pmod{2}$. Let $N$ be the composite of prime factors so that

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k},$$

(5)

where $k > 1$ and $\alpha_i \in \mathbb{Z^+}$. The algorithm will be able to factorize the integer $N$. We can also assume that $N$ is not a prime power, i.e. $k > 1$ and that there exists at least one $p_i \neq p_j$. A prime power can be found with a known classical method in polynomial time. Let us choose a $x \in \mathbb{Z}_N$ randomly, where $\mathbb{Z}_N = \{1, 2, \ldots, N - 1, N\}$. The next step is to use Euclid’s algorithm which determines the greatest common divisor. If $x$ and $N$ are not relatively prime, then we will find a factor by using Euclid’s algorithm. A factor is equal
to $\gcd(x,N)$ if $\gcd(x,N) \neq 1$. If $x$ and $N$ are relatively prime, the task will be to find the order of $x$ in the group $\mathbb{Z}_N$. The algorithm will search for the smallest $r \in \mathbb{Z}^+$ so that $x^r \equiv 1 \pmod{N}$; consequently $r$ is called order $r$ of $x$. If

$$x^r \equiv 1 \pmod{N} \quad (6)$$

and $r$ is an even integer, then it is possible to factorize $x^r - 1$ as

$$x^r - 1 = (x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1). \quad (7)$$

The integer $N$ will share at least one factor with $(x^{\frac{r}{2}} - 1)$ or $(x^{\frac{r}{2}} + 1)$ since $N$ divides $x^r - 1$. These factors can be calculated by Euclid’s algorithm. Let us presume that an even $r$ has been found, which gives us the possibility to determine the factors equal to $\gcd(x^{\frac{r}{2}} + 1, N)$ and $\gcd(x^{\frac{r}{2}} - 1, N)$. There is nevertheless an obvious exclusion that $x^{\frac{r}{2}} \equiv 1 \pmod{N}$ due to the definition of order $r$. However, it may occur that $x^{\frac{r}{2}} \equiv -1 \pmod{N}$, i.e. only trivial factors will be found since $N$ is the greatest common divisor to $x^{\frac{r}{2}} + 1$ and $N$. There will be two factors, $\gcd(x^{\frac{r}{2}} + 1, N)$ and $\gcd(x^{\frac{r}{2}} - 1, N)$, in case the order $r$ of $x$ is an even number and $x^{\frac{r}{2}} \neq -1 \pmod{N}$.

\section*{The Quantum Fourier Transform}

The discrete Fourier transform maps a vector $V_1 = \alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ to another vector $V_2 = \beta_0, \beta_1, \ldots, \beta_{n-1}$, where $\alpha, \beta \in \mathbb{C}$. In traditional mathematics the discrete Fourier transform is defined as follow:

$$V_2 = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \alpha_j e^{2\pi i j k/n}. \quad (8)$$

The quantum Fourier transform (QFT) is similar to the discrete Fourier transform, yet QFT will be represented in computational symbols. The QFT is a function which acts on $q = \log_2 n$ qubits.

\textbf{Definition 1.} \textit{The quantum Fourier transform is a function which maps basis states to a linear combination of basis states}

$$\text{QFT}[e[j]] = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{2\pi i j k/n} e[k]. \quad (9)$$

\textit{The quantum Fourier transform for an arbitrary state $\psi$ is}

$$\text{QFT} \left\{ \sum_{j=0}^{n-1} \alpha_j e[j] \right\} = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \alpha_k e^{2\pi i j k/n} e[k]. \quad (10)$$

The decomposable of QFT is used to match quantum gates.

\textbf{Lemma 1.}

$$\text{QFT}[e[j]] = \frac{1}{\sqrt{n}} (e[0] + e^{\frac{2\pi i}{n}} e[1]) \otimes (e[0] + e^{\frac{2\pi i}{n}} e[1]) \otimes \cdots \otimes (e[0] + e^{\frac{2\pi i}{n}} e[1]). \quad (11)$$
We will construct this decomposition by using the Rotation, the Hadamard and the CNOT gates. Every arbitrary unitary operator may be represented by combinations of single qubit gates and CNOT gates [5]. QFT can be expressed in single quantum gates as:

\[
\text{Swap}(H_q(R_q,q-1H_q-1(\cdots (R_q,2\cdots R_{3,2}H_2(R_q,1\cdots R_{2,1}H_1|e[u])))))
\]

where swap is a combination of \(3q\) numbers of CNOT gates. This decomposition of QFT requires \(q\) operations on the first qubit and \(q-1\) operations on the second qubit, and so on. Hence it follows that the decomposition needs \(\frac{1}{2}(q+1)q\ H\) and \(R_k\) operations. To obtain the right order, we swap the decomposition; thus we need \(3q/2\) or \(3(q-1)/2\) more operations. Altogether, the decomposition of QFT is require in the order of \(q^2\) gates e.i. QFT uses \(O(q^2)\) elementary operations.

QUANTUM COMPUTATION AND SHOR’S FACTORING ALGORITHM.

Shor’s algorithm can be executed in four steps. First let us choose an arbitrary integer \(x \in \mathbb{Z}^+\) which will be smaller than the integer \(N\) that we want to factorize. The second step is to make sure that the chosen integer is not a prime factor. If it is a prime factor the algorithm can result in the prime factor and stop the algorithm. It is only in the third step we need to implement the algorithms in a quantum computer. At this stage we will use the quantum Fourier transform in the algorithm to find the order \(r\) of \(x\). Restart the algorithm if \(r\) is odd or \(x^r \equiv -1 \pmod{N}\). In the last stage a classical computer will calculate the factors and output gcd \((x^r \pm 1, N)\). Let us start the algorithm with an \(N\) and choose \(n = 2^q\) so that \(N^2 \leq n \leq 2N^2\). It will prepare two registers \(|e[0]\rangle|e[0]\rangle\) with \(q\)-qubits in the quantum computer.

1. Let us set up the first register in superposition

\[
\frac{1}{\sqrt{n}} \sum_{c=0}^{n-1} e[c] |e[0]\rangle.
\]

(13)

2. Let us then compute \(x^c \pmod{N}\) in the second register and the computer will be in state

\[
\frac{1}{\sqrt{n}} \sum_{c=0}^{n-1} e[c] |e[x^c \pmod{N}]\rangle.
\]

(14)

3. Next, we need to compute the QFT on the first register \(e[c]\)

\[
QFT |e[c]\rangle = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{2\pi i ck/n} |e[k]\rangle.
\]

(15)

The machine will be in state

\[
\psi = \frac{1}{\sqrt{n}} \sum_{c=0}^{n-1} \sum_{k=0}^{n-1} e^{2\pi i ck/n} |e[k]\rangle |e[x^c \pmod{N}]\rangle.
\]

(16)
Now let us measure the first register in the quantum machine for any \( k \) in \( e[k] \). Order \( r \) of \( x \) can be found as a denominator of one of the convergents to \( k/n \), where the probability to find \( r \) depends on the number of qubits. To find the order \( r \), we need to apply continued fractions to the \( k/n \)

### SHOR’S ALGORITHM IN THE MATHEMATICA CODE

This chapter will introduce a Mathematica program code which implements Shor’s algorithm in a classical computer. We will follow the Mathematica program code evolutions and compare this with Shor’s algorithm. This comparison will demonstrate a connection between the classical computer and the quantum computer. The program will try to find two factors to \( N \), where \( N \) is an odd prime factorization and has at least two different prime factors.

\[
N = 3 * 5; q = \{\text{Log}[2, N^2]\};
\]

```mathematica
Do[x = Random[Integer, {2, N - 1}];
If[\text{GCD}[x, N] == 1, SecondStep; QFT; OutPrint
, Print[\text{"Chosen x=", x, \text{\textquotesingle a \textquotesc{multiplier of }, GCD[x, N], \text{\textquotesc{.}}\";}];
, \left(\frac{160 \text{\text{Log}[Log}[N]]}{9}\right)]
```

The algorithm will choose \( q = \lfloor \text{Log}_2(N^2) \rfloor \) so that the algorithm will find a factor with large probability, i.e. if it selects \( q \) to satisfy \( N^2 \leq 2^q < 2N^2 \), the two factors will then be found with a probability of at least \( \frac{9}{160 \text{\text{Log}[Log}[N]]} \). The program will choose a random integer \( x \in \{2, 3, \ldots, N - 1\} \).

```mathematica
<<\text{NumberTheory\textquoteleft\text{ContinuedFractions\textquoteleft}}e[a\_, x\_, b\_] := a e[x, b] 
\text{e}[x\_, x\_, y\_] := x a e[x, y] + \beta e[y, x];
\text{O}_{i, j} \{v\} := \text{Chop[Expand[v / (e[x\_] \rightarrow \text{ReplacePart[e[x], e[(x[i]] / 0, 1)]]}]}
\text{O}_{i, j} \{v\} := \text{Chop[Expand[v / (e[x\_] \rightarrow (e[(x[i], 1 \text{\text{\text{]] \rightarrow 0 / (e[\alpha, \beta] \rightarrow e[Sequence@@Take\{x, i - 1\}, 1, \alpha, 
Sequence@@Take\{x, (i + 1, j - 1)\}, 
Sequence@@Take\{x, (j + 1, -1)\}]})])]
R := \{e[0] \rightarrow \frac{1}{\sqrt{2}} (e[0] + e[1]), e[1] \rightarrow \frac{1}{\sqrt{2}} (e[0] - e[1])\}
R := \{e[1, 1] \rightarrow e[\alpha] e[1, 1]\}
\text{Swap} := \{e[i, j] \rightarrow e[j, i]\}
```

We will only use the computational basis states \( e[x_1, \ldots, x_d] \), where \( x_j \in \{0, 1\} \). The commands \( e[a\_, x\_, y\_] := a e[x, y] \) and \( e[a, \xi (a e[x] + \beta e[y]), b] := \xi a e[x, b] + \xi \beta e[y, b] \) will give the program code a connection to the tensor product. The command \( \text{O}_{i, j} \{v\} \) is a one-qubit operator which takes vector \( v \) as an attribute and operates with \( O \) on the qubit in position \( i \). Likewise, the command \( \text{O}_{i, j} \{v\} \) is a
two-qubits operator which takes vector \( v \) as an input and operates with \( O \) on the qubit in position \( i \) and \( j \). To compute QFT the algorithm requires three gates, Hadamard (\( H \)), Rotation (\( R \)) and Swap (\( \text{Swap} \)).

FirstStep[\( q_-, x_-, N_\)] := Expand\[ \frac{1}{\sqrt{2^q}} \sum_{c=0}^{2^q-1} e[\text{Sequence}@@\text{IntegerDigits}[c, 2, q], \text{Sequence}@@\text{IntegerDigits}[0, 2, q]] \]

SecondStep := \{FirstStep[\( q_-, x_-, N_\)] := Expand\[ \frac{1}{\sqrt{2^q}} \sum_{c=0}^{2^q-1} e[\text{Sequence}@@\text{IntegerDigits}[c, 2, q], \text{Sequence}@@\text{IntegerDigits}[\text{Mod}[x^c, N], 2, q]] \]; \( u = \text{FirstStep}[q, x, N] \} \}

The command FirstStep prepares the first register in a superposition. Since the first step is pointless in a classical computer; consequently it will be excluded in the code and we go directly to the second step in the code. SecondStep calculates \( x^c \) (mod \( N \)) in the second register, where \( q \) is the number of qubits.

QFT := \( \{(\text{For}[i = 1, i \leq q, i++, u = H[i] \text{ \_} u; \text{For}[j = i + 1, j \leq q, j++, u = R_i,j \text{ \_} u])\} \)

For\[i = 1, i \leq \text{IntegerPart}[\frac{q}{2}], i++, u = \text{Swap}_{i,q-1} \text{ \_} u; \text{OutQFT} = (u \text{ \_} \text{a} \_ e[y_] \_ b \_ e[y_] \Rightarrow \text{Together}[\text{Abs}[a^2], e[y]]; \text{Probability} = \text{List}@@\text{OutQFT} \_ \text{a} \_ e[y_] \Rightarrow \text{Abs}[a^2], e[y]); \text{Probability} = \text{Probability} / \_ \_ e[y_] \Rightarrow a; b = \{\text{Probability}[1]\}; \text{For}[i = 1, i \leq \text{Length}[\text{Probability}], i++, b = \text{AppendTo}[b, b[i-1] \_ \text{Probability}[i]]]; \}

For\[r = \text{Random}[]; \text{For}[i = 1, i \leq \text{Length}[b], i++, \text{If}[r \leq b[i], \text{MeasureQFSTest} = i; \text{Break[]}]; p = \{\text{List}@@\text{OutQFT} \_ \_ e[x_] \Rightarrow \text{FromDigits}[\{\text{Sequence}@@\text{Take}[\{x\}, 2]\}]) \}

The QFT will act on the state \( u \) by means of the three gates in the following order: \( H_q(R_{q,q-1}H_{q-1}(\cdots(R_{q,2}\cdots(R_{q,1}H_1)H_2)H_3(\cdots)))). \) The third line in the program code will swap the qubits. All terms with identical computational basis states will be collected in the command OutQFT. Probability is a list of the probabilities used to measure one of the terms in the register. One of the terms will be randomly chosen taking into consideration of probability to measure the state. The position of the chosen term will be saved in MeasureQFSTest. Finally, the list \( p \) of decimal numbers is derived from the binary list OutQFT.

OutPrint := \{\text{CFD} := \text{Denominator}\[\text{Convergents}[\text{\_p[MeasureQFSTest]} /\_ 2^q] \]

Do\[\text{If}[\text{Mod}[x^{\text{CFD}[1]}, N] = 1 \&\& \text{EvenQ}[\text{CFD}[1]] \&\& \text{Mod}[x^{\text{CFD}[1]}, N] \_ N-1, \text{Print}[^\text{Factors} \text{a}_1=", \text{GCD}[N, x^{\text{CFD}[1]} + 1], \text{\ and\ "} a_2=", \text{GCD}[N, x^{\text{CFD}[1]} - 1], \text{\ have been found.}^\text{\_}], \{i, 1, \text{\_Length}[\text{CFD}]\}] \]
The randomly chosen value in the register is in \( p[[\text{MeasureQFTStep}]] \). In CFD the program saves the denominator of convergents \( p[[\text{MeasureQFTStep}]]/2^q \). From this we can select all even denominators, where \( x^{CDF} \equiv 1 \pmod{N} \) and \( x^{CDF} \neq N - 1 \pmod{N} \). If any of the denominators satisfies these three conditions, it will give us two factors.

**CONCLUSION**

In this study we have constructed a computational language for simulations of quantum algorithms and presented the program code for an algorithm. We have also demonstrated that every unitary operator has a representation in this computational language. An important future challenge is to develop this computational language to include the theory of density operator.

**ACKNOWLEDGMENTS**

I would like to thank my supervisor Prof. Andrei Khrennikov for fruitful discussions on the foundations of quantum computing. I am also grateful to Yaroslav Volovich for his involvement and ideas in this research.

**REFERENCES**

Paper VII

Simulation of Deutsch-Jozsa algorithm in mathematica.

Peter Nyman
Simulation of Deutsch-Jozsa Algorithm in Mathematica

Peter Nyman

International Center for Mathematical Modeling
in Physics, Engineering and Cognitive science
MSI, Växjö University, S-35195, Sweden

Abstract. This study examines the simulation of Deutsch-Jozsa algorithm in Mathematica. The program code implemented on a classical computer will be a straight connection between the mathematical formulation of quantum mechanics and computational methods in Mathematica. This program code will be a foundation of a universal simulation language.

Keywords: Deutsch-Jozsa algorithm, quantum computing, Mathematica

PACS: 03.67.Lx, 02.70.-c, 07.05.Tp

INTRODUCTION

A general quantum simulation language on a classical computer will open the opportunity to compare an experiential result from the developments of quantum computers with the mathematical theory. Our aim is therefore to construct a general program code where it will be easy to implement algorithms. A couple of examples of various methods for the simulation of quantum algorithms were given in [1, 2, 3]. We have in previous research [4] studied Shor’s algorithm [5] and implemented this algorithm in the high-level language Mathematica. A simulation of a quantum algorithm on classical computers will give us the possibility to compare the results of quantum computers with the output of the physically more stable classical computers. In the development of quantum algorithms it will be interesting to test new algorithms on a classical computer. This article will describe the connection between future quantum computers and today’s simulations of quantum computers. Thus, the computational language will include the quantum mechanical terminology such as quantum operators and quantum states. This simulation language will include the most essential operations in quantum computing. Examples of these operations will be the Hadamard, Controlled not and Toffoli gates. In earlier studies (see [6]) we have introduced a framework in a computational language constructed on the formulation of quantum mechanics. In this study we will go further and use this framework when we implement Deutsch-Jozsa algorithm as an example of a quantum algorithm. The aim is to construct a computational language and describe a straightforward connection between Dirac’s mathematical formulation of quantum mechanics and our program code. However, the mathematically described algorithms will have a clear mathematical structure even after that we have implemented this algorithms as a program code. Moreover a program with mathematical structure will give us short program codes and this simulation of Deutsch-Jozsa algorithm will use only few lines of program code. This simulation is a link from the mathematical theory of quantum algorithms to its implementation on a quantum computer. Let us begin with a demonstration of the framework we have constructed in the high-level program language Mathematica, this will probably be used for future algorithms.

THE SIMULATION FRAMEWORK

This section we will introduce a framework constructed for the simulation of quantum algorithms on classical computers. We will point out that there is a symbolic similarity between our framework and the mathematical framework. This framework will be a computational dual to Dirac’s bra-ket notation. A quantum state in n dimensions can be represented by a linear combination of n numbers of basis vectors. In the two-dimensional case a quantum state \(|\phi\rangle\) is represented as a superposition of two basis vectors, say \(|0\rangle\) and \(|1\rangle\), known as computational basis (computational basis, see [7, 8]). In this basis a quantum state \(|\phi\rangle\) is represented as

\[|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,\]  (1)
where \(\alpha\) and \(\beta\) are complex numbers such as \(|\alpha|^2 + |\beta|^2 = 1\). We will introduce some new symbols for the states of the computational basis as follows: \(e[0] = |0\rangle\) and \(e[1] = |1\rangle\). This is the foundation for the structure of the program code. For more than one-qubit we will use the computational basis states \(e[x_1, \ldots, x_n] = |x_1, \ldots, x_n\rangle\), where \(x_j \in \{0, 1\}\) or by using the more compact notation \(e[y] = |y\rangle\), where \(y = x_02^n + \cdots + x_12^{n-1}\). We will, write the state \(\phi\) as \(e[\phi] = \alpha e[0] + \beta e[1]\), by analogy to (1). The operator \(A\) acts on the state \(\phi\) and is often written as \(A[\phi]\) in the quantum mechanical literature. To match these symbols, we will use the computational symbols \(A[e[\phi]]\) for this operation. A computational problem will be that the computer must regard expressions equal if they have identical meaning even if these notations are not identical. As an example the expression \(e[0, e[1], 1]\) must be equal to \(e[0, 1, 1]\) in the code. We can bring in the command \(e[0, e[1], 1] := e[0, 1, 1]\) or the more general \(e[a\ldots, e[b\ldots], e]\) := \(e[a, b, c]\) to solve this problem. Moreover, the program code must be able to handle the linearity of the tensor product. Let \(e[\cdot]\) be vectors and \(\alpha\) a complex number. We define the tensor product as

\[
\alpha(e[v] \otimes e[w]) = (\alpha e[v]) \otimes e[w] = e[v] \otimes (\alpha e[w]), \tag{2}
\]

\[
(e[v] + e[w]) \otimes e[x] = e[v] \otimes e[x] + e[w] \otimes e[x], \tag{3}
\]

\[
e[v] \otimes (e[w] + e[x]) = e[v] \otimes e[w] + e[v] \otimes e[x]. \tag{4}
\]

We add two commands to the program code that will implement this definition of the tensor product. The command

\[
e[a\ldots, \alpha, e[x\ldots], b\ldots] := \alpha \cdot e[a, \ldots, x, \ldots, b]\]

will transform \(e[a] \otimes e[x] \otimes e[c]\) to \(e[a \otimes x \otimes b] = e[a, x, b]\). This command is the computational dual to the tensor expression in Dirac's notation \(|a\rangle \otimes |\alpha\rangle \otimes |b\rangle = |\alpha a x b\rangle\). The other command

\[
e[a\ldots, \xi, e[x\ldots], \beta, e[y\ldots], b\ldots] := \xi \cdot e[a, \ldots, x, \ldots, y, \ldots, b] + \beta \cdot e[a, \ldots, y, \ldots, b]\]

will transform \(e[a] \otimes \xi (e[x] + \beta e[y]) \otimes e[b]\) to \(\xi e[a, x, b] + \beta e[a, y, b]\). Let \(U\) be an arbitrary one-qubit quantum gate. Then \(U\) will transform an arbitrary state \(e[\phi]\) which is represented in the computational basis states as \(e[\phi] = a e[0] + b e[1]\) to the state \(U[e[\phi]] \rightarrow a (c_1 e[0] + c_2 e[1]) + b (c_3 e[0] + c_4 e[1])\), where \(a, b, c, d\) are complex numbers. We add the Mathematica gate \(U\) to the program code as follows \(U[e[0]] \rightarrow c_1 e[0] + c_2 e[1]\) and \(U[e[1]] \rightarrow c_3 e[0] + c_4 e[1]\). For example, the Hadamard gate \(H\) will be added in Mathematica as the command \(H := \{e[0] \rightarrow 1/\sqrt{2}(|e[0] + e[1]|), e[1] \rightarrow 1/\sqrt{2}(|e[0] - e[1]|)\}\). We will define a one-qubit gate \(O_i\) as an operator which acts on the qubit in position \(i\) and leaves the other qubits unchanged. The program code must be able to operate with a gate on an arbitrary qubit. Consequently we will define an operator \(O_i\) in the Mathematica code. Defined the operator \(O_i\), then \(O_i\) is a function of \(O_i[e[v]] \rightarrow e[y]\). Similarly, we will define \(O_{ij}\) as an operator which operates as the two-qubits operator on the qubits in positions \(i, j\) and leaves the other qubits unchanged. Now we have the tools to build quantum circuits.

\section*{An Introduction to the Deutsch-Jozsa Algorithm}

The function \(f: \{0, 1\}^n \rightarrow \{0, 1\}\) is constant if all output is 0 or 1 for all input. The function \(f\) is balanced if half of all outputs are 0. The Deutsch-Jozsa problem was introduced by David Deutsch and Richard Jozsa with the task to decide the function \(f\) property, which will be constant or balanced (see [9]). The most naive classical algorithm will need \(2^{n-1} + 1\) outputs to decide the property of the function. Deutsch and Jozsa showed that the Deutsch-Jozsa algorithm would solve this task with only one output. In the implementation of Deutsch-Jozsa algorithm we will use to the unitary operator \(U_f\) which we defined as

\[
U_f: |x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle
\]

where \(\oplus\) is the binary addition modulo 2. We will use the function \(U_f := e[i\ldots, j\ldots] := e[i, \text{Mod}[j + f[i], 2]]\) to apply this operator in our simulation language. Let us describe Deutsch-Jozsa problem in the following example, Bob choose a function with one of the two properties, constant or balanced, and Alice task is to decide its property. To her delight she will get Bob to calculate this function in a quantum computer. Bob prepare the register in superposition before he operates with the operator \(U_f\) that will act on all qubits at once. Before Bob measure the register and send the result to Alice he will apply the Hadamard gate to the \(n - 1\) first qubits. If Bob sends \(0^\otimes m\) to Alice then she will be sure that his function was constant, otherwise she will make a conclusion that the function was balanced.
THE DEUTSCH-JOZSA ALGORITHM IN A SYMBOLIC LANGUAGE

Let us use our simulation language to help Alice solve her problem to decide the functions property. You may follow the algorithm in the circuit in figure 1. We start on the left with the register \( e[0,\ldots,0,1] \) and operate Hadamard gate \( H := \{ e[0] \rightarrow \frac{1}{\sqrt{2}} (e[0] + e[1]), e[1] \rightarrow \frac{1}{\sqrt{2}} (e[0] - e[1]) \} \) on this state which will place all qubits in superposition

\[
\psi_1 = \frac{1}{\sqrt{2^n+1}} \sum_{x \in \{0,1\}^n} e[x](e[0] - e[1]).
\]

At this point we will use the operator \( U_f \) and the register will be in the state \( \psi_2 \)

\[
\psi_2 = \frac{1}{\sqrt{2^n+1}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} e[x](e[0] - e[1]).
\]

Before measurement the algorithm will apply the Hadamard operation on all qubits except the last one and

\[
\psi_3 = \frac{1}{2^{n+1/2}} \sum_{z \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot z + f(x)} e[z](e[0] - e[1]),
\]

where \( x \cdot z = x_1 z_1 + x_2 z_2 + \cdots + x_n z_n \mod 2 \). Measurement of the first \( n \) will give the output \( e[0,\ldots,0] \) if and only if \( f \) is constant, otherwise it is balanced. This will follow from the fact that the probability to measure \( e[0,\ldots,0] \) is

\[
\frac{1}{2^n} \left| \sum_{x \in \{0,1\}^n} (-1)^{x \cdot 0 + f(x)} \right|^2
\]

and it will be one if \( f(x) = 0 \) or \( f(x) = 1 \) for all \( x \) where \( \emptyset = \{0,0,\ldots,0\} \). Moreover, it will be zero probability to measure \( e[0,\ldots,0] \) if \( f \) is balanced since \( (5) \) will be equal to zero.

THE DEUTSCH-JOZSA ALGORITHM IN MATHEMATICA

The Deutsch-Jozsa algorithm is implemented as follow. We will begin to define characteristic properties of quantum computers in the program code.

Listing 1: Definition of register and quantum gates in mathematica.

```mathematica
e[a___, α_. e[x___], b___] := α e[a, x, b]
e[a___, β_. e[y___], b___] := β e[a, y, b]
O_{f, i} |v_. := Chop[Expand[v /. (e[x___] ⇒ ReplacePart[e[x], e[{x}[[i]]] / O, i)])]
O_{f, i} |v_. := Chop[Expand[v /. O]]
H := \{ e[0] := 1/√2 (e[0] + e[1]), e[1] := 1/√2 (e[0] - e[1]) \}
Uf := \{ e[i___, j_] := e[i, Mod[j + f[i], 2]] \}
Enlarge[v_, i_] := v /. i \{ x___ :\} := e[Sequence@@Table[0, {i - 1}], x]
```

The essential points of this framework is linearity and the tensor products. We have described before how tensor products and a register will be represented in our simulation language. The two-first lines (see listing 1) are computationally powerful and they will manage to implement some of the essential properties in simulation as linearity, superposition...
and tensor products. Line 3-4 in the same listing will define the operator mapping one or two qubits respectively. The Deutsch-Jozsa algorithm only need to apply the two operators $H$ and $U_f$. The last line will enlarge a state of the one qubit $e[1]$ to the $n$-qubit state $e[0,0,0,\ldots,1]$. The operators, linearity, superposition and tensor products are now defined. The implementation of the algorithm will follow from the listing 2.

Listing 2: Deutsch-Jozsa algorithm in mathematica.

```mathematica
q = 8;
ϕ = Enlarge[e[1], q];
Do[ϕ = (H_i | ϕ), {i, q}];
ϕ = U_{f}^{q-1,q} | ϕ;
ϕ = ϕ /. {f[1, x__] → 0, f[0, x__] → 1};
Do[ϕ = (H_i | ϕ), {i, q - 1}];
ϕ
```

There are certain advantages to compare this listing 2 with the circuit (1). By using the command `Enlarge` on state $e[1]$, defined in line 2, will the register will be prepared in the $q$-qubit state $e[0,0,0,\ldots,1]$. In listing 2 on line 3-4 the Hadamard operator and the unitary function are applied on all the qubits in the register. In line 5 we will be able to choose the function property. In this example we have chosen a balanced function. The algorithm will output the result when the Hadamard operator have been applied on the $q-1$ first qubits. The output will contain zeros in $q-1$ first qubits if and only if $f$ is constant, otherwise it is balanced.

CONCLUSION

In this study we have constructed a computational language for simulations of quantum algorithms and presented a program code for Deutsch-Jozsa algorithm. We have also demonstrated a general framework for simulation of quantum computers on classical computers. An important future challenge is to develop this computational language to include the most of the well-known quantum algorithms.

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Andrei Khrennikov for fruitful discussions on the foundations of quantum computing. I am also grateful to Yaroslav Volovich for his involvement and ideas in this research.

REFERENCES

Simulation of Simon’s algorithm in mathematica.

Peter Nyman
Simulation of Simon’s Algorithm in Mathematica

Peter Nyman
International Center for Mathematical Modeling in Physics, Engineering and Cognitive science, MSI, Växjö University, S-35195, Sweden

Abstract
A general quantum simulation language on a classical computer provides the opportunity to compare an experiential result from the development of quantum computers with mathematical theory. The intention of this research is to develop a program language that is able to make simulations of quantum mechanical processes as well as quantum algorithms. This study examines the simulation of quantum algorithms on a classical computer with a symbolic programming language. We use the language Mathematica to make a simulation of well-known quantum algorithms. The program code implemented on a classical computer will be a straight connection between the mathematical formulation of quantum mechanics and computational methods. This gives us an uncomplicated and clear language for implementations of algorithms. The computational language includes essential formulations such as quantum state, superposition and quantum operator. This symbolic programming language provides a universal framework for examining the existing as well as future quantum algorithms. This study contributes with an implementation of a quantum algorithm in a program code where the substance is applicable in other simulation of quantum algorithms.

keywords: Mathematica, Simon’s Algorithm, Quantum Algorithm Simulation, Quantum Computing

1 Introduction
Our aim is to construct a general program code where it will be easy to implement algorithms. A couple of examples of various methods for the sim-
ulation of quantum algorithms were given in the papers [1, 2, 3, 4]. We have in previous research [5] studied Shor's algorithm [6] and implemented this algorithm in the high-level language Mathematica. A simulation of a quantum algorithm on classical computers will give us the possibility to compare the results of quantum computers with the output of the physically more stable classical computers. In the development of quantum algorithms it will be interesting to test new algorithms on a classical computer. This article will describe the connection between future quantum computers and today's simulations of quantum computers. Thus, the computational language will include the quantum mechanical terminology such as quantum operators and quantum states. This simulation language will include the most essential operations in quantum computing. Examples of these operations will be the Hadamard, Controlled not and Toffoli gates. In earlier studies (see [7]) we have introduced a framework in a computational language constructed on the formulation of quantum mechanics. In this study we will go further and use this framework when we implement Simon's algorithm as an example of a quantum algorithm. The aim is to construct a computational language and describe a straightforward connection between Dirac's mathematical formulation of quantum mechanics and our program code. However, the mathematically described algorithms will have a clear mathematical structure even after that we have implemented this algorithms as a program code. Moreover a program with mathematical structure will give us short program codes and this simulation of Simon's algorithm will use only few lines of program code. This simulation is a link from the mathematical theory of quantum algorithms to its implementation on a quantum computer. Let us begin with a demonstration of the framework we have constructed in the high-level program language Mathematica, this will probably be used for future algorithms.

2 The Simulation Framework

This section we will introduce a framework constructed for the simulation of quantum algorithms on classical computers. We will point out that there is a symbolic similarity between our framework and the mathematical framework. This framework will be a computational dual to Dirac's bra-ket notation. A quantum state in \( n \) dimensions can be represented by a linear combination of \( n \) numbers of basis vectors. In the two-dimensional case a quantum state \( |\phi\rangle \) is represented as a superposition of two basis vectors, say \( |0\rangle \) and \( |1\rangle \), known as computational basis (computational basis, see [8, 9]). In this basis
a quantum state $|\phi\rangle$ is represented as

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,$$  \hspace{1cm} (1)

where $\alpha$ and $\beta$ are complex numbers such as $|\alpha|^2 + |\beta|^2 = 1$. We will introduce some new symbols for the states of the computational basis as follows: $e[0] = |0\rangle$ and $e[1] = |1\rangle$. This is the foundation for the structure of the program code. For more than one-qubit we will use the computational basis states $e[x_1, \ldots, x_n] = |x_1 \ldots x_n\rangle$, where $x_j \in \{0, 1\}$ or by using the more compact notation $e[y] = |y\rangle$, where $y = \sum x_j 2^{n-j}$. We will, write the state $\phi$ as $e[\phi] = \alpha e[0] + \beta e[1]$, by analogy to (1). The operator $A$ acts on the state $\phi$ and is often written as $A|\phi\rangle$ in the quantum mechanical literature. To match these symbols, we will use the computational symbols $e[0], e[1]$ and is often written as $A|\phi\rangle$ for this operation. A computational problem will be that the computer must regard expressions equal if they have identical meaning even if these notations are not identical. As an example the expression $e[0, e[1], 1]$ must be equal to $e[0, 1, 1]$ in the code. We can bring in the command $e[0, e[1], 1] := e[0, 1, 1]$ or the more general $e[a,\_\_ e[b,\_\_], c,\_\_] := e[a,b,c]$ to solve this problem. Moreover, the program code must be able to handle the linearity of the tensor product. Let $e[\_\_ \_]$ be vectors and $a$ a complex number. We define the tensor product as

$$\alpha(e[v] \otimes e[w]) = (\alpha e[v]) \otimes e[w] = e[v] \otimes (\alpha e[w])$$  \hspace{1cm} (2)

$$e[v_1] + e[v_2] \otimes e[w] = e[v_1] \otimes e[w] + e[v_2] \otimes e[w]$$  \hspace{1cm} (3)

$$e[v] \otimes (e[w_1] + e[w_2]) = e[v] \otimes e[w_1] + e[v] \otimes e[w_2].$$  \hspace{1cm} (4)

We add two commands to the program code that will implement this definition of the tensor product. The command

$$e[a,\_\_, \alpha \cdot e[x,\_\_], b,\_\_] := \alpha e[a, x, b]$$

will transform $e[a] \otimes \alpha e[x] \otimes e[c]$ to $\alpha e[a \otimes x \otimes b] = \alpha e[a, x, b]$. This command is the computational dual to the tensor expression in Dirac’s notation $|a\rangle \otimes \alpha |x\rangle \otimes |b\rangle = \alpha |a x b\rangle$. The other command

$$e[a,\_\_, \xi \cdot (\alpha \cdot e[x,\_] + \beta \cdot e[y,\_]), b,\_\_] := \xi \alpha e[a, x, b] + \xi \beta e[a, y, b]$$

will transform $e[a] \otimes \xi (\alpha e[x] + \beta e[y]) \otimes e[b]$ to $\xi e[a] \otimes e[x, b] + \xi e[a, y, b]$. Let $U$ be an arbitrary one-qubit quantum gate. Then $U$ will transform an arbitrary state $e[\phi]$ which is represented in the computational basis states as $e[\phi] = a e[0] + b e[1]$ to the state $e[\phi] \rightarrow a(c_1 e[0] + c_2 e[1]) + b(c_3 e[0] + c_4 e[1])$, where $a, b, c_i$ are complex numbers. We add the Mathematica gate $U$ to the program code as follows $U[e[0]] \rightarrow c_1 e[0] + c_2 e[1] and U[e[1]] \rightarrow c_3 e[0] + c_4 e[1]$. For example, the Hadamard gate $H$ will be added in Mathematica as the
command $H:=\{ e[0] \rightarrow 1/\sqrt{2}(e[0] + e[1]), \ e[1] \rightarrow 1/\sqrt{2}(e[0] - e[1])\}$. We will define a one-qubit gate $O_i$ as an operator which acts on the qubit in position $i$ and leaves the other qubits unchanged. The program code must be able to operate with a gate on an arbitrary qubit. Consequently we will define an operator $O_i$ in the Mathematica code. Defined the operator $O_i$ as $O_i = I^{\otimes i-1} \otimes U \otimes I^{\otimes n-i}$ which acts on $n$-qubits, where $I$ is the one-qubit unit operator and $U$ is an arbitrary one-qubit operator. Then operator $O_i$ is a function of $O_i|e[v] \rightarrow e[v]$. Similarly, we will define $O_{i,j}$ as an operator which operates as the two-qubits operator on the qubits in positions $i,j$ and leaves the other qubits unchanged. Now we have the tools to build quantum circuits.

3 An Introduction To Simon’s Algorithm

Daniel Simon proposed following problem[10, 11]; Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a 2 to 1 function and let

$$f(x) = f(y) \iff x = y \oplus a,$$

where $\oplus$ is the binary addition modulo 2. The task will be to decide the value $a \in \{0, 1\}^n$ i.e. we will try to find the period $a$. In the implementation of Simon’s algorithm we will use to the unitary operator $U_f$ which we defined as

$$U_f : |x^\otimes n\rangle |y^\otimes n\rangle = |x^\otimes n\rangle |y^\otimes n\rangle f(x).$$

This operator $U_f$ will act on the state in a quantum black box i.e. we have no knowledge about the properties of the function $f$. Let us describe Simon’s problem in the following example, Bob choose a function that is periodic in respect of binary addition modulo 2. Alice task will be to decide this period $a$. To her delight she will get Bob to calculate this function in a quantum computer. Bob prepare two $n$-qubit registers. He apply the Hadamard gate on the first register so that it will be in superposition and the then he apply the $U_f$ gate on the both register. Now bob will have two alternative, he measure the the last $n$ qubits and then apply the Hadamard gate to the $n$ remaining qubits or he apply Hadamard gate to the $n$ first qubits and after that he measure them. The both methods will give the same results, but we reduce the expressions if we think that makes a measurement before the last operations. Bob will now measure the quantum computer and he will get some $y_1 \in \{0, 1\}^n$ so that $y_1 \cdot a = 0$. He will now restart the algorithm and hope that he measures a different value $y_2$. He will continue restarting the algorithm until he measured $n$ different values $y_i$. The next task is to solve
a linear equation so that $y_i \cdot a = 0$ for all $i \in \{1, 2, \ldots, n\}$

$$|0\rangle^\otimes n \xrightarrow{H} |H\rangle^\otimes n \xrightarrow{Uf} |H\rangle^\otimes n \xrightarrow{H}$$

Figure 1. The Simon’s algorithm circuit

4 The Simon’s Algorithm in a Symbolic Language

Let us use our simulation language to decide the period $a$ of a periodic function i.e. we search for an $a$ so that $f(x) = f(y) \iff x = y \oplus a$. Prepare two registers with the states $e[0]^\otimes n$ and then begin with the algorithm, which can be followed in figure 1. The first operations will be to act with the Hadamard gate on the first register. The registers will now be in superposition of the states $e[x] e[0]^\otimes n$ so that

$$\psi_1 = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} e[x] e[0]^\otimes n.$$ 

In this step the Uf operator act on the both registers and then the registers will be in the states;

$$\psi_2 = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} e[x] e[f(x)].$$

Measure the second register for some state $f(x_0)$ where $x_0 \in \{0,1\}^n$. All the state $f(x)$ will have equal probability $2^{-n}$ to be measured. Moreover, since $f(x_0) = f(x_0 + a)$ are the first register in the states

$$\psi_3 = \frac{1}{\sqrt{2}} \left(e[x_0] + e[x_0 \oplus a]\right),$$

after measurement of the second register. Then apply Hadamard to $\psi_3$ and use that $H e[x] = \sum_y (-1)^{x \cdot y} e[y]$ to get

$$\psi_4 = \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a) \cdot y}\right) e[y]$$

or equivalent

$$\psi_4 = \frac{1}{\sqrt{2^{n-1}}} \sum_{y : a \cdot y = 0} (-1)^{x_0 \cdot y} e[y].$$
Measurement of the first register will give a $y_0 \in \{0, 1\}^n \otimes n$ where $y_0 \cdot a = 0$. Now restart the algorithm and measure a new value $y_1$ with the probability $1 - 2/(2^{n-1})$ that satisfy $y_1 \neq y_0$ and $y_1 \neq 0$, in that case this gives a equation $y_1 \cdot a = 0$ where $y_1$ linearly independent to $y_0$. We will repeat this to we obtain $n - 1$ numbers of linearly independent equations $y_0 \cdot a = 0, y_1 \cdot a = 0, \ldots, y_{(n-2)} \cdot a = 0$ to solve. In generality will the probability be $1 - 2^{1+i-n}$ for $i < n$ to measure a $y_i$ so that $y_0 \cdot a = 0, y_1 \cdot a = 0, \ldots, y_{(i-1)} \cdot a = 0$ is linearly independent. Solve the equations to find $a$.

5 The Simon’s Algorithm in Mathematica

The Simon’s algorithm is implemented as in Mathematica as following. The First part of the program code is a framework for simulation of quantum algorithms in Mathematica and this part define characteristic properties of quantum computers.

Listing 1: Definition of register and quantum gates in Mathematica.

1. $e[a\_,\alpha\_.e[x\_],b\_\_] := \alpha e[a,x,b]$
2. $e[a\_,\xi\_.(\alpha\_.e[x\_]+\beta\_.e[y\_]),b\_\_] := \xi\alpha e[a,x,b]+\xi\beta e[a,y,b]$
3. $O_i|v_\_:=\text{Chop[Expand[v/.(e[x\_]:}→\text{ReplacePart[e[x],e[[x]][[i]]]/.0,0}])]]$
4. $O_{i,j}|v_\_:=\text{Chop[Expand[v/.0}])]}$
5. $H:=\{e[0]:→1/\sqrt{2}(e[0]+e[1]),e[1]:→1/\sqrt{2}(e[0]-e[1])\}$
6. $Uf := \{e[i\_,\_\_, j]\_\_→e[e[x]][[1;\_;i]],$
7. $e[Sequence @@Mod[List@e[x][[i+1;;j]]]$
8. $+f[Sequence@e[x][[1;;i]],2]]\}$
9. $Enlarge[\psi\_,i\_] :=$
10. $\psi/.\psi→(\psi/.e[x\_]:→ e[Sequence@@Table[0, \{i-1\}, x])$

One of the characteristic properties we need to define in the program code is the tensor product for the qubit states and the operators. Another property is the linearity of the operators. We have described before how superposition, tensor products and the linear operator will be represented in our simulation language. The two-first lines (see listing 1) are computationally powerful and they will manage to implement some of the essential properties in simulation as linearity, superposition and tensor products. Line 3–5 in the same listing will define the operator mapping one or two qubits respectively. The Simon’s
algorithm only need to apply the two operators $H$ and $U_f$. The last line will enlarge a state of the one qubit $e[0]$ to the $n$-qubit state $e[0] \otimes e[0]$. The operators, linearity, superposition and tensor products are now defined. The implementation of the Simon’s algorithm will follow from the listing 2.

Listing 2: Simon’s algorithm in mathematica.

```mathematica
q = 5; Measure = {};
While[Length[Union[Measure]] \[LessEqual] q,
  \[Phi] = Enlarge[e[0], q];
  Do[\[Phi] = (H_i | \[Phi]), \{i, q\}];
  a = \{1, 0, 1, 1, 0\};
  f[0, x___] := Mod[{0, x}, 2];
  f[1, x___] := Mod[{1, x} + a, 2];
  \[Phi] = Uf_q, 2 \[Phi];
  Do[\[Phi] = (H_i | \[Phi]), \{i, q\}];
  Probability = List@@Expand[\[Phi] /. \[Alpha]_. e[y___] -> Abs[\[Alpha]]^2;
  Table[Probability[[i+1]] = Plus@@Take[Probability, {i, i+1}],
    \{i, 1, Length[\[Phi]]-1\}];
  r = Random[];
  AppendTo[Measure, Take[\[Phi][[1+LengthWhile[Probability, \# < r &]], 2]], \{1, q\}]]
]

Out[11]=\{e[0, 0, 0, 0, 1], e[0, 0, 1, 1, 0], e[0, 1, 0, 0, 1],
  e[0, 1, 1, 0, 0], e[1, 0, 1, 0, 0], e[1, 0, 1, 0, 1]\}
```

There are certain advantages to compare this listing 2 with the circuit in figure 1. Numbers of qubits in the two registers will be selected and an empty list of measured states creates in the first line. The algorithm iterates with use of a `while` loop until it has measured $q$-number linearly independent states. The command `Enlarge` applied on state $e[0]$, defined in line 3, will prepare the two register in $q$-qubit states $e[0] \otimes e[0]$. In line 4 the Hadamard operator are applied to the first register. Take the unknown period $a$ to be $\{1, 0, 1, 1, 0\}$ in this example. In this specific example will the function be defined as in line 6 and 7, here must mention that it is conceivable too explicit defined every function value. After that the code will apply $U_f$ the
both registers where the first register is the control qubits and the second is targets qubits. Then apply the Hadamard gate to the first register before measurement. Moreover will a measurement of the first register means that one state of all the $q$-qubit states in superposition will be randomly chosen where the measurement probability is equal to square of the absolute value of the phase. Consequently, must the simulation of a measurement depend on the phase. As a first step to make a measurement will line 10 to 12 create a list called Propability, with contains the probability to measure the states. The algorithm chose a randomly $r \in [0, 1]$. This random $r$ decides which of the element (state) that will be measured in the list Propability. The while loop in the final line will measure an element and add it to the list Measure. Finally the latest line will output all measured elements in the list Measure. It remains to solve the linear equation to find $a$, but since it will be done in a classical computer will we leave this besides. We can easy verify that $y \cdot a = 0$ for all measured states in the output.

6 Results

The program have been tested for a number different periods that require 2 to 6 qubits registers (see table 1 for the result of some of the test). The runtime for 6 qubit registers indicates to be around 300 second. The result from the test with more than 6 qubits showed that this simulation of Simon’s algorithm will give a long runtime larger registers.

7 conclusion

We have constructed a computational language for simulations of quantum algorithms with are presented by implementation of Simon’s algorithm. This demonstrates a part of a general framework for simulation of quantum computers on classical computers. The test shows that this simulation is not effective for registers with more than 6 qubits. An important future challenge is to continue to develop this computational language to include all well-known quantum algorithms.
<table>
<thead>
<tr>
<th>Period</th>
<th>Output</th>
<th>Runtime/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>{0, 0, 1}</td>
<td>{e[0,0,0],e[0,1,0],e[1,0,0],e[1,1,0]}</td>
<td>0.125</td>
</tr>
<tr>
<td>{0, 1, 0}</td>
<td>{e[0,0,0],e[0,0,1],e[1,0,0],e[1,0,1]}</td>
<td>0.157</td>
</tr>
<tr>
<td>{1, 1, 1}</td>
<td>{e[0,0,0],e[1,0,1],e[1,1,0],e[1,1,1]}</td>
<td>0.187</td>
</tr>
<tr>
<td>{0, 1, 1, 0}</td>
<td>{e[0,0,0,0],e[0,0,0,1],e[1,0,0,0],e[1,0,0,1],e[0,1,1,0]}</td>
<td>0.375</td>
</tr>
<tr>
<td>{0, 0, 1, 0}</td>
<td>{e[0,0,0,0],e[0,0,0,1],e[0,1,0,0],e[1,0,0,1],e[1,1,0,0],e[1,1,0,1]}</td>
<td>0.437</td>
</tr>
<tr>
<td>{1, 0, 1, 0}</td>
<td>{e[0,0,0,0],e[0,0,0,1],e[0,1,0,0],e[1,0,1,0],e[1,1,0,0],e[1,1,1,0]}</td>
<td>0.313</td>
</tr>
<tr>
<td>{0, 1, 1, 0, 0}</td>
<td>{e[0,0,0,1,1],e[0,1,1,0,1],e[1,0,0,1,0],e[1,1,0,1,0],e[1,1,1,0,0],e[1,1,1,1,0]}</td>
<td>5.203</td>
</tr>
<tr>
<td>{1, 0, 1, 1, 0}</td>
<td>{e[0,0,0,0,1],e[0,1,1,0,0],e[0,1,1,1,1],e[1,0,0,1,0],e[1,1,0,1,0],e[1,1,1,0,1]}</td>
<td>7.344</td>
</tr>
<tr>
<td>{1, 1, 1, 0, 0}</td>
<td>{e[0,1,0,0,1],e[0,1,1,0,1],e[0,1,1,1,0],e[1,0,1,0,1],e[1,1,0,1,0],e[1,1,1,0,0],e[1,1,1,0,1]}</td>
<td>3.157</td>
</tr>
<tr>
<td>{1, 0, 1, 1, 0, 0}</td>
<td>{e[0,0,0,0,0,1],e[0,0,0,0,1],e[0,0,0,1,1],e[0,0,1,1,1,0],e[0,1,0,1,1,0],e[1,0,0,1,1,0],e[1,1,0,1,0,0],e[1,1,1,0,1,0]}</td>
<td>215.265</td>
</tr>
<tr>
<td>{1, 1, 1, 1, 0, 0}</td>
<td>{e[0,0,0,0,1,1],e[0,0,1,1,0,1],e[0,1,0,1,1,0],e[0,1,1,0,1,1],e[1,1,0,0,0,1],e[1,1,1,0,0,1],e[1,1,1,0,1,0],e[1,1,1,1,0,0],e[1,1,1,1,1,0],e[1,1,1,1,1,1]}</td>
<td>306.938</td>
</tr>
</tbody>
</table>

Table 1: Test results

8 acknowledgments

I would like to thank my supervisor Professor Andrei Khrennikov for fruitful discussions on the foundations of quantum computing. I am also grateful to Yaroslav Volovich for his involvement and ideas in the beginning of this research.

References


A compact program code for simulations of quantum algorithms in
A compact program code for simulations of quantum algorithms in classical computers.

Peter Nyman
A compact program code for simulations of quantum algorithms in classical computers

Peter Nyman
International Center for Mathematical Modeling in Physics, Engineering and Cognitive science, MSI, Växjö University, S-35195, Sweden

June 4, 2008

Abstract
A general quantum simulation language on a classical computer provides the opportunity to compare an experiential result from the development of quantum computers with mathematical theory. The intention of this research is to develop a program language that is able to make simulations of all quantum algorithms in same framework. This study examines the simulation of quantum algorithms on a classical computer with a symbolic programming language. We use the language Mathematica to make simulations of well-known quantum algorithms. The program code implemented on a classical computer will be a straight connection between the mathematical formulation of quantum mechanics and computational methods. This gives us an uncomplicated and clear language for the implementations of algorithms. The computational language includes essential formulations such as quantum state, superposition and quantum operator. This symbolic programming language provides a universal framework for examining the existing as well as future quantum algorithms. This study contributes with an implementation of a quantum algorithm in a program code where the substance is applicable in other simulations of quantum algorithms.

1 Introduction
A simulation of a quantum algorithm on physically more stable classical computers will give material for comparison to unstable quantum computers
output. A simulation of the algorithm will be a natural step in the development of new quantum algorithms. This article will describe the connection between future quantum computers and today’s simulations of quantum computers. Thus, the computational language will include the quantum mechanical terminology such as quantum operators, quantum states and tensor product. This simulation language will include the most essential operations in quantum computing and have a design making it straightforward to add all sorts of gates. Examples of these operations will be the Hadamard, pauli matrices gates and the oracle. We have in previous study use the same framework for implementation of Simon’s Algorithm, Deutsch-Jozsa algorithm, Shor’s algorithm and error correction [1, 10, 11, 12], this studies is presented in [7, 8, 9]. We will now present an expansion of this framework to include an implementation of Grover’s search algorithm [3]. Grover’s search algorithm will as the previous studied algorithms have a compact and mathematical notation in this program. Quantum operators will be represented as currently literature [2, 4, 6] usually represent quantum operators. Using these compact and mathematical representations will we be able to implement Grover’s search algorithm in an only a few numbers of lines short program code. We will emphasize that all quantum algorithms implemented in this framework [7, 8, 9] will make use of only a few numbers of lines in the program code. The aim is to construct a computational language and describe a straightforward connection between Dirac’s mathematical formulation of quantum mechanics and our program code. However, the mathematically described algorithms will have a clear mathematical structure even after that we have implemented this algorithms as a program code. This simulation is a link from the mathematical theory of quantum algorithms to its implementation on a quantum computer. We must point out that this simulation of these algorithms in a classical computer will obviously not speedup execute time. Let us begin with a demonstration this framework constructed in the high-level program language Mathematica that can be used for other algorithms without larger modifications.

2 The Simulation Framework

This section we will introduce a framework constructed for the simulation of quantum algorithms on classical computers. We will point out that there is a symbolic similarity between our framework and the mathematical framework. This framework will be a computational dual to Dirac’s bra-ket notation. A quantum state in n dimensions can be represented by a linear combination of n numbers of basis vectors. In the two-dimensional case a quantum state
$|\phi\rangle$ is represented as a superposition of two basis vectors, say $|0\rangle$ and $|1\rangle$, known as computational basis (computational basis, see [5, 6]). In this basis a quantum state $|\phi\rangle$ is represented as

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

(1)

where $\alpha$ and $\beta$ are complex numbers such as $|\alpha|^2 + |\beta|^2 = 1$. We will introduce some new symbols for the states of the computational basis as follows: $e[0] = |0\rangle$ and $e[1] = |1\rangle$. This is the foundation for the structure of the program code. For more than one-qubit we will use the computational basis states $e[x_1, \ldots, x_n] = |x_1 \ldots x_n\rangle$, where $x_j \in \{0, 1\}$ or by using the more compact notation $e[y] = |y\rangle$, where $y = x_n2^n + \cdots + x_12^{n-1}$. We will, write the state $\phi$ as $e[\phi] = \alpha e[0] + \beta e[1]$, by analogy to (1). The operator $A$ acts on the state $\phi$ and is usually written as $A|\phi\rangle$ in the quantum mechanical literature. To match these symbols, we will use the computational symbols $A|e[\phi]\rangle$ for this operation. A trick will be to regard $|x_1 \ldots y_1 \ldots y_m \ldots x_n\rangle$ as $|x_1 \ldots y_1 \ldots y_m \ldots x_n\rangle$ in order to simplify the program code. This will be a computational problem since Mathematica will distinguish between $e[x_1, \ldots, e[y_1, \ldots, y_m, \ldots, x_n]]$ and $e[x_1, \ldots, y_1, \ldots, y_m, \ldots, x_n]$. The computer must regard these expressions as equally even if the notations are not identical equal to each other. As an example the expression $e[0, e[1], 1]$ must be equal to $e[0, 1, 1]$ in the code. We can bring in the command $e[0, e[1, 1]] := e[0, 1, 1]$ or the more general $e[a___, e[x__], b___] := e[a, b, c]$ to solve this problem. Moreover, the program code must be able to handle the linearity of the tensor product. Let $e[.]$ be vectors and $\alpha$ a complex number. We define the tensor product as

$$\alpha(e[v] \otimes e[w]) = (\alpha e[v]) \otimes e[w] = e[v] \otimes (\alpha e[w]),$$

(2)

$$e[v_1] + e[v_2]) \otimes e[w] = e[v_1] \otimes e[w] + e[v_2] \otimes e[w].$$

(3)

$$e[v] \otimes (e[w_1] + e[w_2]) = e[v] \otimes e[w_1] + e[v] \otimes e[w_2].$$

(4)

Two short commands in the program code that will implement this definition of the tensor product. The command

$$e[a___, \alpha_. e[x__], b___] := \alpha e[a, x, b]$$

will transform $e[a] \otimes \alpha e[x] \otimes e[c]$ to $\alpha e[a \otimes x \otimes b] = \alpha e[a, x, b]$. This command is the computational dual to the tensor expression in Dirac’s notation $|a\rangle \otimes \alpha|x\rangle \otimes |b\rangle = \alpha|a \otimes x \otimes b\rangle$. The other command

$$e[a___, \xi_. (\alpha_. e[x__] + \beta_. e[y__]), b___] :=$$

$$\xi \alpha e[a, x, b] + \xi \beta e[a, y, b].$$
will transform $e[a] \otimes \xi(\alpha e[x] + \beta e[y]) \otimes e[b]$ to $\xi\alpha e[a,x,b] + \xi\beta e[a,y,b]$. Let $U$ be an arbitrary unitary one-qubit quantum gate. Then $U$ will transform an one-qubit state $e[\phi]$ which is represented in the computational basis states as $e[\phi] = a e[0] + b e[1]$ to the state $U|e[\phi]\rangle \rightarrow a(c_1 e[0] + c_2 e[1]) + b(c_3 e[0] + c_4 e[1])$, where $a, b, c_i$ are complex numbers. We add the Mathematica gate $U$ to the program code as follows $U|e[0]\rangle \rightarrow c_1 e[0] + c_2 e[1]$ and $U|e[1]\rangle \rightarrow c_3 e[0] + c_4 e[1]$. For example, the Hadamard gate $H$ will be added in Mathematica as the command $H := \{ e[0] \rightarrow 1/\sqrt{2}(e[0] + e[1]), e[1] \rightarrow 1/\sqrt{2}(e[0] - e[1]) \}$. We will define a one-qubit gate $O_i$ as an operator which acts on the qubit in position $i$ and leaves the other qubits unchanged. The program code must be able to operate with a gate on an arbitrary qubit. Consequently we will define an operator $O_i$ in the Mathematica code. Defined the operator $O_i$ as $O_i = I^\otimes i^{-1} \otimes U \otimes I^\otimes n-i$ which acts on $n$-qubits, where $I$ is the one-qubit unit operator and $U$ is an arbitrary one-qubit operator. Then operator $O_i$ is a function of $O_i|e[\psi]\rangle \rightarrow e[\psi]$. Similarly, we will define $O_{i,j}$ as an operator which operates as the two-qubits operator on the qubits in positions $i, j$ and leaves the other qubits unchanged. Now we have the tools to build the quantum circuit for quantum algorithms and error correction.

2.1 Introduction to Grover’s search algorithm

In developments of this framework have a number of quantum algorithms been implemented [7, 8, 9] in purpose to examine its applicability, moreover will this study be a presentation of Grover’s algorithm in this code. Let us present the idea of Grover’s search algorithm [3] by an example. Imagine that you have a list of $N$-numbers of registration numbers and theirs car owners sort in alphabetical order by theirs family name. You see a registration number and want to check the name of the owner to this car. Since the list is in alphabetical order you may need to check all $N$-numbers of car owner. The average number you need to check to find the owner when you search by registration number will be $N/2 = O(N)$ (this is the classical linear search), but with use of Grover’s search algorithm will you success in $O(\sqrt{N})$ time. Suppose we will search for the owner of the car with registration number “DKG 313”. Let us say that “DKG 313” is in the unknown place $54$ that we have decode as $x_0 = e[110111]$. We define a function

$$f(x) = \begin{cases} 1 & \text{for } x = x_0, \\ 0 & \text{for } x \neq x_0. \end{cases}$$

Let us assume that it possible to build a unitary operator (oracle) such that

$$U_f e[x] e[y] = e[x] e[y \oplus f(x)],$$

where $\oplus$ is the XOR gate.
where $\oplus$ means binary addition modulo 2. Prepare an initial state $e[\phi_0] = e[x]e[1]$ and apply the Hadamard gate $H$ on the last qubit $e[\phi_1] = I \otimes H e[\phi_0] = e[x]e[-]$ where $e[-] = \frac{1}{\sqrt{2}}(e[0] - e[1])$. Apply $U_f$ on this state

$$U_f e[\phi_1] = e[x] \frac{1}{\sqrt{2}} (e[0 \oplus f(x)] - e[1 \oplus f(x)]) = (-1)^{f(x)} e[x] e[-]$$

and a phase sign shift will mark if the $x$ is the searched element. If $U_f$ acts on a superposition $H^{\otimes_{q+1}} e[0]^{\otimes q} e[1]$ as

$$U_f(H^{\otimes_{q+1}} e[0]^{\otimes q} e[1]) = \frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^{\otimes q}} U_f e[i] e[-]$$

$$= \frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^{\otimes q}} (-1)^{f(i)} e[i] e[-]$$

$$= \frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^{\otimes q}} (e[i] - 2 e[x_0]) e[-]$$

will the searched element $x_0$ be mark in the superposition with a phase sign shift. Since the absolute value in square of the corresponding phase still is unchanged will the possible to measure this element is unchanged even if the search element is marked by sign sift. For Grover’s search algorithm will we need one more quantum gate $U_0$ which is a special case of $U_f$ where $x_0 = 0$ such that

$$f(x) = \begin{cases} 
1 & \text{for } x = 0, \\
0 & \text{for } x \neq 0.
\end{cases}$$

A short description of algorithm will follow (for details see [6]). Prepare the quantum computer in the initial state $e[\psi_0] = e[0]^{\otimes q} e[1]$ and apply the Hadamard gate $e[\psi_1] = H^{q+1} e[\psi_0]$ to all the qubits (see figure 1). All the qubits will now be in superposition. Continue with the Grover iteration first operations which is to apply $U_f$ on this state $e[\psi_2] = U_f e[\psi_1]$, thus will the last qubit will flip the sign to phase of the searched element. The search element is now marked, but the probability to measure this element is unchanged. Apply the Hadamard gate $e[\psi_3] = H^q e[\psi_2]$ to all the qubits without the last one. The third and fourth operations in the Grover iteration are $U_0$ and $H^q$, after that $e[\psi_5] = H^q U_0 e[\psi_3]$. Repeat the Grover iteration if it is necessarily otherwise measurement the state $e[\psi_6] = \text{Measure} e[\psi_5]$. There is easy to show [6] that we obtain the largest probability to measure $x_0$ if the Grover iteration is repeated for

$$k = \text{round} \left( \frac{\arccos(\sqrt{M/N})}{\arcsin(2\sqrt{M(N-M)/N})} \right)$$

where $M$ is the number of elements in the database.
numbers of times where \( N \) is numbers of elements in the search list and \( M \) is numbers of solutions to the problem.

\begin{align*}
\left| 0 \right>^{\otimes q} & \xrightarrow{H^{\otimes q+1}} \left| U_f \right> \xrightarrow{H^{\otimes q}} \left| U_0 \right> \xrightarrow{H^{\otimes q}} \left| 1 \right>
\end{align*}

Figure 1. The Grover’s algorithm circuit using one Grover iteration

### 2.2 Grover algorithm in Mathematica

Prepare the quantum computer in the initial state \( e[0]^{\otimes q+1} \) by use of the command in line 1. Apply \( X \) to the qubit number \( q + 1 \) and \( H^{\otimes q+1} \) to all qubits to get to the state

\[
\frac{1}{\sqrt{N}} \sum_{y \in \{0,1\}^q} e[y] e[-].
\]

After applying \( U_f \) to this state will the searched element be marked in the state

\[
\left( \frac{1}{\sqrt{N}} \sum_{y \in \{0,1\}^q} e[y] - \frac{2}{\sqrt{N}} e[x_0] \right) e[-].
\]

Next step in the algorithm will be to apply the sequence of operators \( H^{\otimes q} U_0 H^{\otimes q} \). In order to attain the maximum probability to success must the Grover iteration \( G^{\otimes q+1} = U_f H^{\otimes q} U_0 H^{\otimes q} \) be repeated for \( k \) times. After the first Grover iteration will the state be

\[
\left( \frac{N - 4}{N - \sqrt{N}} \sum_{y \in \{0,1\}^q} e[y] + \frac{2}{\sqrt{N}} e[x_0] \right) e[-]
\]

where the probability of a measurement will be given by

\[
\begin{cases}
  p(\text{Measure } y \neq x_0) = \left| \frac{N-4}{N} \frac{1}{\sqrt{N}} \right|^2 = |\cos 3\theta|^2,
  \\
  p(\text{Measure } y = x_0) = \left| \frac{N-4}{N} \frac{1}{\sqrt{N}} + \frac{2}{\sqrt{N}} \right|^2 = |\sin 3\theta|^2.
\end{cases}
\]
where \( \sin \theta = \frac{1}{\sqrt{N}} \). For a general point of view consider \( k \) numbers of Grover iteration, then the probability measure \( x_0 \) will be written

\[
\begin{align*}
    p(\text{Measure } y \neq x_0) &= |\cos(2k + 1)\theta|^2, \\
    p(\text{Measure } y = x_0) &= |\sin(2k + 1)\theta|^2.
\end{align*}
\]

The positive integer \( k \) that minimize \( |1 - |\sin(2k + 1)\theta|^2| \) will give the maximum probability to measure \( x_0 \), this is equivalent to

\[
k = \text{round}\left( \frac{\arccos\left(\frac{1}{\sqrt{N}}\right)}{2 \arcsin\left(\frac{1}{\sqrt{N}}\right)} \right). \tag{7}
\]

Let \( M = 1 \) (i.e. only one element will match the search problem) then we see that the equation (7) is equivalent to (6), using the fact that \( 2 \arcsin\left(\frac{1}{\sqrt{N}}\right) = \arcsin\left(\sqrt{N - \frac{1}{N}}\right) \).

### 2.3 Grover's search algorithm in the Simulation Framework

The Grover’s search algorithm algorithm is implemented as in Mathematica as following. The First part (see the four first code lines in listing 1) of the program code is a framework for simulation of quantum algorithms in Mathematica and this part define characteristic properties of quantum computers. The follow code lines in the listing 1 defined the operators \( X, H, U_f \) and \( U_0 \).

Notice that it is straightforward to introduce new gates in this program from the mathematical representation of gates.

Listing 1: Definition of register and quantum gates in mathematica.

\[
\begin{align*}
\text{InitialState}[q_] &= \text{e}\left[\text{Sequence}\left[\text{Table}\left[0,\{q\}\right]\right]\right] \\
\text{e}[a___,\alpha_.\text{e}[x___],b___] &= \alpha\text{e}[a,x,b] \\
\text{e}[a___,\xi_.(\alpha_.\text{e}[x___]+\beta_.\text{e}[y___]),b___] &= \xi\alpha\text{e}[a,x,b]+\xi\beta\text{e}[a,y,b] \\
O_{\text{ij}}&:=\text{Chop}\left[\text{Expand}\left[\text{v}./.\left(\text{e}[x___]\rightarrow\text{ReplacePart}\left[\text{e}[x],\text{e}\left[\{x\}\left[\text{ii}\right]\right]/.0,i\right]\right)\right]\right] \\
0_{\text{ji}}&:=\text{Chop}\left[\text{Expand}\left[\text{v}./.0\right]\right] \\
X &= \{\text{e}[0]\rightarrow\text{e}[1],\text{e}[1]\rightarrow\text{e}[0]\} \\
H &= \{\text{e}[0]\rightarrow 1/\sqrt{2}(\text{e}[0]+\text{e}[1]),\text{e}[1]\rightarrow 1/\sqrt{2}(\text{e}[0]-\text{e}[1])\} \\
Uf[i___,j___] &= \{\text{e}[x___]\rightarrow\text{e}[e[x][[1;;i]]],
\end{align*}
\]

7
Let us choose an example to describe the part of the program which represents the quantum circuit for Grover algorithm. Assume that we will search for \( x_0 \) in a list of 120 element where \( x_0 \) have four solutions 19, 29, 39, 79 (i.e. \( M = 4 \)).

The register needs to contain \( q + 1 \) numbers of qubits where \( q = \log_2(N) = 7 \) and \( N = 128 \geq 120 \). In this special case will numbers of iterations be

\[
k = \text{round} \left( \frac{\arccos \left( \sqrt{\frac{4}{128}} \right)}{2 \arcsin \left( \sqrt{\frac{4 (128 - 4)}{128}} \right)} \right) = 4.
\]

In next part the register will be prepared in the initial state \( e^{0 \otimes q+1} \), then algorithm will follow the circuit in figure 1 and apply the \( X \) gate and the other gates. The last part of the program will simulate a measurement of the register where probability to measure a basis state depends on its corresponding phase.

Listing 2: A search in a list of 120 elements.

```plaintext
Clear [f, g]
q = Length [f @@ IntegerDigits [120, 2]]
f @@ IntegerDigits [79, 2, q] := 1;
f @@ IntegerDigits [39, 2, q] := 1;
f @@ IntegerDigits [19, 2, q] := 1;
f @@ IntegerDigits [29, 2, q] := 1; f[x_] := 0;
g @@ IntegerDigits [0, 2, q] := 1; g[x_] := 0;
M = 4; N = 2^q;
K[M_, N_] :=
Round [ArcCos [Sqrt [M/N]]/(ArcSin [2 Sqrt [M (N-M)/N]])];
Φ = InitialState [q+1];
Φ = (H[q+1] | Φ);
Do [Φ = (H[i] | Φ), {i, q+1}];
Do [Φ = (Uf[q,q+1] | Φ); 8
```
\[
\text{Do } [\Phi = (H_i | \Phi), \{i, q\}] ;
\]
\[
\Phi = (U_{0,q+1} | \Phi);
\]
\[
\text{Do } [\Phi = -(H_i | \Phi), \{i, q\}, \{j, K(M,N)\}] ;
\]
\[
\Phi = (H_{q+1} | \Phi);
\]
\[
\text{Probability} = \text{List@@Expand}[\Phi]/. \alpha_\_ \_ e[y___] \rightarrow \text{Abs}[\alpha]^2;
\]
\[
\text{Table}[\text{Probability}[[i+1]] = \text{Plus@@Take}[\text{Probability},\{i,i+1\}],\{i,1,\text{Length}[\Phi]-1\};
\]
\[
r = \text{Random[]} ;
\]
\[
\text{Take}[\Phi[[1+\text{LengthWhile}[\text{Probability},#<r\&],-1]]]
\]

References


Paper X

Simulation of quantum error correcting code.

Peter Nyman
Simulation of Quantum Error Correcting Code

Peter Nyman
International Center for Mathematical Modeling in Physics, Engineering and Cognitive science, MSI, Växjö University, S-35195, Sweden

September 21, 2008

Abstract
This study considers implementations of error correction in a simulation language on a classical computer. Error correction will be necessarily in quantum computing and quantum information. We will give some examples of the implementations of some error correction codes. These implementations will be made in a more general quantum simulation language on a classical computer in the language Mathematica. The intention of this research is to develop a programming language that is able to make simulations of all quantum algorithms and error corrections in the same framework. The program code implemented on a classical computer will provide a connection between the mathematical formulation of quantum mechanics and computational methods. This gives us a clear uncomplicated language for the implementations of algorithms.

1 Introduction
The mathematical model of quantum computers is an idealization of a physical quantum computer. In a physical quantum computer decoherence with the environment causes errors. The use of error correction provides a possibility to reduce the effect of errors. A number of mathematical models in error correction in the form of an error-correcting code have been developed. We will implement some of them in the simulation framework developed by us. This framework in Mathematica is a computer language for the simulation of quantum computers in classical computers. Within this framework we will transform the mathematical model of quantum mechanics into a computational code. Thus it will be a straightforward matter to implement quantum
algorithms and error correcting codes in this language. More specifically this means that it will represent the Dirac notation and theory connected to this notation in a natural manner. We build a state in superposition of the computational basis |0⟩ and |1⟩ act on this state with quantum gates. This state will be an increased n-qubit state with the use of the tensor product. Thus the n-qubit state will be a superposition of the computational basis |0⟩⊗n to |1⟩⊗n. We will act with one qubit gate or two-qubit controlled-NOT gates on this state. Together this will give us a sufficient device for simulating quantum computers.

2 The Simulation Framework

Let us introduce the part of the program that will be the framework for the simulation of error correction. This framework must naturally be part of the quantum algorithm that demands protection from error correction. In a previous study several well-known quantum algorithms have been implemented in this framework by the author (see [5, 4, 3]). We will point out that there is a symbolic similarity between our framework and the mathematical foundation of quantum computing. For this reason we will represent the code by a simple modification of Dirac’s notation. A quantum state in n dimensions can be represented by a linear combination of n numbers of basis vectors as { e[0], e[1], ..., e[n] } = { e[0]⊗n, e[0]⊗n−1 ⊗ e[1], ..., e[n]}. In the two-dimensional case a quantum state |φ⟩ is represented as a superposition of two basis vectors, say |0⟩ and |1⟩, known as computational basis (computational basis, see [1, 2]). In this basis a quantum state |φ⟩ is represented as

|φ⟩ = α|0⟩ + β|1⟩, (1)

where α and β are complex numbers such as |α|² + |β|² = 1. We will introduce some new symbols for the states of the computational basis as follows: e[0] = |0⟩ and e[1] = |1⟩. This is the foundation for the structure of the program code. For more than one qubit we will use the computational basis states e[ x₁, ..., xₙ ] = |x₁ ... xₙ⟩, where x_j ∈ {0, 1} or the more compact notation e[y] = |y⟩, where y = xₙ2₀ + ... + x₁2ⁿ⁻¹. We will write the state φ as e[φ] = αe[0] + βe[1], in analogy to (1). The operator A acts on the state φ and is usually written as A|φ⟩ in the quantum mechanical literature. To match these symbols we will use the computational symbols A[e[φ]] for this operation. One might regard |x₁ ... y₁ ... y_m ... xₙ⟩ as |x₁ ... y₁ ... y_m ... xₙ⟩ in order to simplify the program code. This will be a computational problem, since Mathematica will distinguish between e[ x₁, ..., e[y₁, ..., y_m], ..., xₙ]
and $e[x_1, \ldots, y_1, \ldots, y_m, \ldots, x_n]$. The computer must regard these expressions as equal even if the notations are not identical with each other. As an example the expression $e[0, e[1], 1]$ must be equal to $e[0, 1, 1]$ in the code. We can bring in the command $e[0, e[1], 1] := e[0, 1, 1]$ or the more general $e[a\_, c[b\_, c\_]] := e[a, b, c]$ to solve this problem. Moreover, the program code must be able to handle the linearity of the tensor product. Let $e[\_]$ be vectors and $\alpha$ a complex number. We define the tensor product as

$$\alpha(e[v] \otimes e[w]) = (\alpha e[v]) \otimes e[w] = e[v] \otimes (\alpha e[w])$$

(2)

$$e[v_1] + e[v_2]) \otimes e[w] = e[v_1] \otimes e[w] + e[v_2] \otimes e[w]$$

(3)

$$e[v] \otimes (e[w_1] + e[w_2]) = e[v] \otimes e[w_1] + e[v] \otimes e[w_2].$$

(4)

Two short commands in the program code will implement this definition of the tensor product. The command

$$e[a\_, , e[x\_1], b\_\_] := e[a, x, b]$$

will transform $e[a] \otimes \alpha e[x] \otimes e[c]$ into $\alpha e[a \otimes x \otimes b] = \alpha e[a, x, b]$. This command is the computational dual to the tensor expression in Dirac’s notation $|a\rangle \otimes \alpha |x\rangle \otimes |b\rangle = \alpha |a x b\rangle$. The other command

$$e[a\_, , \xi\_, (\alpha\_, e[x\_1] + \beta\_, e[y\_1]), b\_\_] := \xi \alpha e[a, x, b] + \xi \beta e[a, y, b]$$

will transform $e[a] \otimes \xi(\alpha e[x] + \beta e[y]) \otimes e[b]$ to $\xi \alpha e[a, x, b] + \xi \beta e[a, y, b]$. Let $U$ be an arbitrary unitary one-qubit quantum gate. Then $U$ will transform a one-qubit state $e[\phi]$, which is represented in the computational basis states as $e[\phi] = a e[0] + b e[1]$, into the state $U|e[\phi] \rightarrow a(c_1 e[0] + c_2 e[1]) + b(c_3 e[0] + c_4 e[1])$, where $a, b, c_i$ are complex numbers. We add the Mathematica gate $U$ to the program code as follows: $U|e[0] \rightarrow c_1 e[0] + c_2 e[1]$ and $U|e[1] \rightarrow c_3 e[0] + c_4 e[1]$. For example, the Hadamard gate $H$ will be added in Mathematica as the command $H := \{e[0] \rightarrow 1/\sqrt{2}(e[0] + e[1]), e[1] \rightarrow 1/\sqrt{2}(e[0] - e[1])\}$. We will define a one-qubit gate $O_i$ as an operator which acts on the qubit in position $i$ and leaves the other qubits unchanged. The program code must be able to operate with a gate on an arbitrary qubit. Consequently, we will define an operator $O_i$ in the Mathematica code. Defined the operator $O_i$ as $O_i = I^{\otimes i-1} \otimes U \otimes I^{\otimes n-i}$, which acts on $n$-qubits where $I$ is the one-qubit unit operator and $U$ is an arbitrary one-qubit operator. Then operator $O_i$ is a function of $O_i|e[v] \rightarrow e[\psi]$. Similarly, we will define $O_{i,j}$ as an operator which operates as the two-qubit operator on the qubits in positions $i, j$ and leaves the other qubits unchanged. Now we have the tools to build the quantum circuit for quantum algorithms and error correction.
2.1 Quantum Error Correcting

Quantum computers need some connection to make them controllable and will therefore never be completely isolated from their surroundings. The quantum computer’s surroundings will influence the quantum computer and cause errors. The effects of errors caused by inaccuracy and decoherence need to be reduced to a minimum. Error correction will help to reduce these effects. Define the 3-qubit ”logical qubits” denoted \(|1_L\rangle\) (logical-one) and \(|0_L\rangle\) (logical-zero), as it is commonly defined in the literature

\[
|0_L\rangle = |000\rangle, \quad |1_L\rangle = |111\rangle.
\]  

Consider some error that will cause a flip on one of the qubits in the logical qubits (e.g. \(|000\rangle\) change to \(|010\rangle\)). This qubit flip will be represented by the Pauli operator \(X_i\), where \(i\) denotes the position of the flipped qubit. Thus we will define \(X_1 = X \otimes I \otimes I\), \(X_2 = I \otimes X \otimes I\) and \(X_2 = I \otimes I \otimes X\). In the same manner a phase flip \(|1\rangle \mapsto -|1\rangle\) and the combination of phase flip and qubit flip will be represented by the Paul operator \(Z_i\) and \(X_iZ_i = -iY_i\), respectively. Moreover, ignore the global phase\(^1\) and denote a combination of a qubit and phase flip as \(Y_i\). A linear combination of the Pauli matrices and the identity matrix will then represent an error operator

\[
E_i = c_1I_i + c_2X_i + c_3Z_i + c_4Y_i.
\]

The use of majority voting and a preparation of the state in logical qubits (5) will protect the state against errors that flip a single qubit. This error code will not protect against a phase flip. The Shor code [6] will overcome this problem. Therefore prepare the state in the logical qubits

\[
|0_L\rangle = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^\otimes 3, \quad |1_L\rangle = \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^\otimes 3.
\]

The encoding of the logical states in Shor’s code is presented in the left part of the circuit in figure 1. Some arbitrary error will be simulated by the error operator 6 in the middle of the circuit. This operator simulates a one-qubit error which can be a phase or a flip error or some combination of these two. The decoding is the inverse to encoding, as will be seen in the circuit.

3 The simulation

The representations of error correcting code (Shor’s code) simulation in Mathematica will follow in this section. First define the quantum computer

\(^{1}\)Two states which are equal up to the global phase will be equal in the observer’s eyes (see: [2, p.93])
This part of the code has defined the register and the quantum gates, and the simulation is ready for the quantum circuit. The next part, which expresses the quantum circuit, is divided into encoding, simulating errors, decoding and measuring. The Enlarge function in the listing 2 will be a simulation of the extension of an arbitrary computational one-qubit state to a nine-qubit
state. The encoding will affect the Shor code, which is easy to compare with the encoding in a quantum circuit 1. Hence read the Mathematica code from inward out. The first quantum gate to implement is the $\text{CN}_{4,1}$ gate, the next one the $\text{CN}_{7,1}$ gate, and so forth. The simulation of noise is implemented by the Error function. After applying some noise we can measure the state and return to the initial state.

Listing 2: Encode and decode

```mathematica
Enlarge[ψ_, i_] := ψ/. e[x] → e[x, Sequence@Table[0, {i}]]
Encoding[ψ_] := CN9,7|CN8,7|CN6,4|CN5,4|CN3,1|H7|H4|H1|CN1,1|ψ)
Error[ψ_, i] := a_i(Id_i|ψ)+b_i(Z_i|ψ)+d_i(Y_i|ψ)
Decoding[ψ_] := T7,4,1|CN7,1|CN4,1|H7|H4|H1|T9,8,7|CN9,7|CN8,7|T6,5,4|CN5,4|T3,2,1|CN3,1|CN2,1|ψ)
Measure[ψ_] := FullSimplify[ψ/. e[y_, x_] → e[y]]
```

Let us encode the arbitrary computational initial state $ψ_0$ and encode by the Shor code. The simulation of a $i$th qubit error on the $ψ_1$ state is implemented by the Enlarge function $[ψ_1,i]$ (specific to this example, an error in qubit 8). Finally, decode and measure the state.

Listing 3: Algorithm

```mathematica
ψ0 = αe[0] + βe[1];
ψ1 =Enlarge[ψ0, 9];
ψ2 = Encoding[ψ1];
ψ3 = Error[ψ2, 8];
ψ4 = Decoding[ψ3];
ψ5 = Measure[ψ4]
```

The output will be $(αe[0] + βe[1]) (a_8 + b_8 + c_8 - id_8)$, where the global phase $|a_8 + b_8 + c_8 - id_8|^2 = 1$ if the error operator is a unitary operator. In fact, the global phase can be ignored and the state $(αe[0] + βe[1]) (a_8 + b_8 + c_8 - id_8)$ and $(αe[0] + βe[1]) (a_8 + b_8 + c_8 - id_8)$ will be considered as equal in an observational point.

References


