Artificial Intelligence 174 (2010) 162–176

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Quantum computation, quantum theory and AI ☆

Mingsheng Ying [a,b,*]

[a] Center of Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
[b] State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, China

ARTICLE INFO

Article history:
Received 30 July 2009
Received in revised form 8 September 2009
Accepted 19 September 2009
Available online 18 November 2009

Keywords:
Quantum computation
Quantum theory
Search
Learning
Discrimination and recognition
Bayesian network
Semantic analysis
Communication

ABSTRACT

The main purpose of this paper is to examine some (potential) applications of quantum computation in AI and to review the interplay between quantum theory and AI. For the readers who are not familiar with quantum computation, a brief introduction to it is provided, and a famous but simple quantum algorithm is introduced so that they can appreciate the power of quantum computation. Also, a (quite personal) survey of quantum computation is presented in order to give the readers a (unbalanced) panorama of the field. The author hopes that this paper will be a useful map for AI researchers who are going to explore further and deeper connections between AI and quantum computation as well as quantum theory although some parts of the map are very rough and other parts are empty, and waiting for the readers to fill in.

1. Introduction

Quantum theory is without any doubt one of the greatest scientific achievements of the 20th century. It provides a uniform framework for the construction of various modern physical theories. After more than 50 years from its inception, quantum theory married with computer science, another great intellectual triumph of the 20th century and the new subject of quantum computation was born.

Quantum computers were first envisaged by Nobel Laureate physicist Feynman [47] in 1982. He conceived that no classical computer could simulate certain quantum phenomena without an exponential slowdown, and so realized that quantum mechanical effects should offer something genuinely new to computation. In 1985, Feynman's ideas were elaborated and formalized by Deutsch in a seminal paper [30] where a quantum Turing machine was described. In particular, Deutsch introduced the technique of quantum parallelism based on the superposition principle in quantum mechanics by which a quantum Turing machine can encode many inputs on the same tape and perform a calculation on all the inputs simultaneously. Furthermore, he proposed that quantum computers might be able to perform certain types of computation that classical computers can only perform very inefficiently.

One of the most striking advances was made by Shor [91] in 1994. By exploring the power of quantum parallelism, he discovered a polynomial-time algorithm on quantum computers for prime factorization of which the best known algorithm

☆ This work was partly supported by the National Natural Science Foundation of China (Grant Nos. 60736011, 60621062) and the National Key Project for Fundamental Research of China (Grant No. 2007CB807901).

* Address for correspondence: Center of Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia.

E-mail address: mying@it.uts.edu.au.

0004-3702/$ – see front matter
doi:10.1016/j.artint.2009.11.009

on classical computers is exponential. In 1996, Grover [52] offered another killer application of quantum computation, and he found a quantum algorithm for searching a single item in an unsorted database in square root of the time it would take on a classical computer. Since database search and prime factorization are central problems in computer science and cryptography, respectively, and the quantum algorithms for them are much faster than the classical ones, Shor and Grover's works stimulated an intensive investigation in quantum computation. Since then, quantum computation has been an extremely exciting and rapidly growing field of research.

Since it revolutionized the very notion of computation, quantum computation forces us to reexamine various branches of computer science, and AI is not an exception. Roughly speaking, AI has two overall goals: (1) engineering goal – to develop intelligent machines; and (2) scientific goal – to understand intelligent behaviors of humans, animals and machines [75]. AI researchers mainly employ computing techniques to achieve both the engineering and scientific goals. Indeed, recently, McCarthy [8] even pointed out that "computational intelligence" is a more suitable name of the subject of AI to highlight the key role played by computers in AI. Naturally, the rapid development of quantum computation leads us to ask the question: how can this new computing technique help us in achieving the goals of AI. It seems obvious that quantum computation will largely contribute to the engineering goal of AI by applying it in various AI systems to speedup the computational process, but it is indeed very difficult to design quantum algorithms for solving certain AI problems that are more efficient than the existing classical algorithms for the same purpose. At this moment, it is also not clear how quantum computation can be used in achieving the scientific goal of AI, and to the best of my knowledge there are no serious research pursuing this problem. Instead, it is surprising that quite a large amount of literature is devoted to applications of quantum theory in AI and vice versa, not through quantum computation. It can be observed from the existing works that due to its inherent probabilistic nature, quantum theory can be connected to numerical AI in a more spontaneous way than to logical AI.

The aim of this paper is two-fold: (1) to give AI researchers a brief introduction and a glimpse of the panorama of quantum computation; and (2) to examine connections between quantum computation, quantum theory and AI. The remainder of the paper is organized as follows: Section 2 is a tutorial of quantum computation for readers who are not familiar with quantum computation and quantum theory. Section 3 surveys some areas of quantum computation which the author is familiar with. Some potential applications of quantum computation in AI are considered in Section 4, and the interplay between quantum theory and AI is discussed in Section 5. A brief conclusion is drawn in Section 6.

2. A tutorial of quantum computation

For convenience of the readers, I will give a very brief introduction to quantum computation in this section. The fundamental principles of quantum theory are embodied very well in the basic apparatus of quantum computation. To illustrate the power of quantum computation, I will present the Deutsch–Jozsa algorithm which I believe to be one of the best examples that a newcomer can appreciate. For more details, we refer to the excellent textbook [74].

2.1. Qubits and quantum registers

The basic data unit in a quantum computer is a qubit, which can be physically realized by a two-level quantum-mechanical system, e.g. the horizontal and vertical polarizations of a photon, or the up and down spins of a single electron. Mathematically, a qubit is represented by a unit vector in the two-dimensional complex Hilbert space, and it can be written in the Dirac notation as follows:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{1}$$

where $|0\rangle$ and $|1\rangle$ are two basis states, and α_0 and α_1 are complex numbers with $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The states $|0\rangle$ and $|1\rangle$ are called computational basis states of qubits. Obviously, they correspond to the two states 0 and 1 of classical bits. The number α_0 and α_1 are called probability amplitudes of the state $|\psi\rangle$. A striking difference between classical bits and qubits is that the latter can be in a superposition of $|0\rangle$ and $|1\rangle$ in the form of Eq. (1). An example state of qubit is: $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

A quantum register is formed by putting multiple qubits together. A state of a quantum register consisting of n qubits is described in the following way:

$$|\psi\rangle = \sum_{t\in\{0,1\}^n} \alpha_t|t\rangle = \sum_{t_1,t_2,\ldots,t_n\in\{0,1\}} \alpha_{t_1t_2\ldots t_n}|t_1t_2\ldots t_n\rangle, \tag{2}$$

where the complex numbers $\alpha_{t_1t_2\ldots t_n}$ are required to satisfy the normalization condition:

$$\sum_{t\in\{0,1\}^n} |\alpha_t|^2 = \sum_{t_1,t_2,\ldots,t_n\in\{0,1\}} |\alpha_{t_1t_2\ldots t_n}|^2 = 1.$$

The state $|\psi\rangle$ in Eq. (2) is a superposition of the computational basis states $|t_1t_2\ldots t_n\rangle$ $(t_1, t_2, \ldots, t_n = 0, 1)$ of the quantum registers. The numbers $\alpha_{t_1t_2\ldots t_n}$'s are the probability amplitudes of $|\psi\rangle$. We can also write:

$$|\psi\rangle = \sum_{t=0}^{2^n-1} \alpha_t|t\rangle$$

if the nonnegative integer $t_1 2^{n-1} + t_2 2^{n-2} + \cdots + t_n 2^0$ is identified with its binary representation $t = t_1 t_2 \ldots t_{n-1}$. Another way to represent the state $|\psi\rangle$ is to write it in the form of column vector:

$$|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{2^n-1} \end{pmatrix}. \tag{3}$$

Several registers can be put together to form a larger register whose state is given in terms of the tensor product of the states of its component registers. Let

$$|\psi_i\rangle = \sum_{t^{(i)}} \alpha_{i,t^{(i)}} \left|t^{(i)}\right\rangle$$

be an n_i qubit state for each $1 \leqslant i \leqslant k$. Then their tensor product is defined to be

$$|\psi_1\rangle \ldots |\psi_k\rangle = \bigotimes_{i=1}^{k} |\psi_i\rangle = \sum_{t^{(1)},\ldots,t^{(k)}} \alpha_{1,t^{(1)}} \ldots \alpha_{k,t^{(k)}} \left|t^{(1)}, \ldots, t^{(k)}\right\rangle.$$

We often simply write $|\psi\rangle^{\otimes k}$ for $\underbrace{|\psi\rangle \ldots |\psi\rangle}_{k}$.

Entanglement is a crucial feature of multiple qubit systems and an extremely useful physical resources in quantum computation and information processing. It is easy to see that there are many $m+n$ qubit states which cannot be written as the tensor product of an m qubit state and an n qubit state. These kind of states are usually called entangled states. An example of two qubit entanglement is the Bell state:

$$|\beta\rangle = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big). \tag{4}$$

2.2. Quantum gates

Typically, quantum computation is realized by quantum circuits consisting of quantum gates. A quantum gate describes a discrete time step of evolution of a closed quantum system. A quantum gate acting on a quantum register consisting of n qubits can be conveniently described by a $2^n \times 2^n$ unitary matrix; that is, a complex matrix U such that $UU^\dagger$ is the identity matrix, where $U^\dagger$ stands for the Hermitian conjugate (or conjugate transpose) of U; that is, the (i, j)-entry of $U^\dagger$ is the complex conjugate of (j, i)-entry of U. If the current state of a quantum register is given by Eq. (3), and

$$U = (u_{ij})_{i,j=0}^{2^n-1}$$

is a quantum gate, then the outcome of performing U on $|\psi\rangle$ is the state

$$|\varphi\rangle = \begin{pmatrix} \beta_0 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{2^n-1} \end{pmatrix} = U|\psi\rangle,$$

where $U|\psi\rangle$ is given according to the usual matrix multiplication; that is,

$$\beta_i = \sum_{j=0}^{2^n-1} u_{ij}\alpha_j$$

for $i = 0, 1, \ldots, 2^n - 1$. One of the most useful single qubit gates is the Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Let U_i be a gate acting on the ith register for each $1 \leqslant i \leqslant k$. Then the tensor product of $U_1, \ldots, U_k$ is a gate acting on the big register formed by the k registers. Formally, it is defined by

$$\left(\bigotimes_{i=1}^{k} U_i\right)\left(\bigotimes_{i=1}^{k} |\psi_i\rangle\right) = \bigotimes_{i=1}^{k} U_i|\psi_i\rangle$$

together with linearity, where $|\psi_i\rangle$ is a state of the ith register for each i. We often write $U^{\otimes k}$ for $\underbrace{U \otimes \cdots \otimes U}_{k}$.

2.3. Quantum measurements

The outcome of quantum computation can only be obtained by measuring certain quantum registers. We only consider quantum measurement in the computational basis. It is well known that quantum measurements in other bases can be carried out by combining unitary transformation and measurement in the computational basis. Suppose we have a quantum register consisting of qubits $q_1, \ldots, q_n$, and $q_{i_1}, \ldots, q_{i_m}$ is a subsequence of $q_1, \ldots, q_n$, where $m \leqslant n$. For any $t \in \{0,1\}^n$, the restriction of t on $q_{i_1}, \ldots, q_{i_m}$ is defined to be the sequence $t|q_{i_1}, \ldots, q_{i_m} = t_{i_1} \ldots t_{i_m} \in \{0,1\}^m$. Let the quantum register be in the state

$$|\psi\rangle = \sum_{t\in\{0,1\}^n} \alpha_t |t\rangle.$$

If a measurement is performed on qubits $q_{i_1}, \ldots, q_{i_m}$, then for each $s \in \{0,1\}^m$, we will get outcome s with probability

$$p(s) = \sum_{t|q_{i_1},\ldots,q_{i_m}=s} |\alpha_t|^2,$$

and the post-measurement state of the quantum register is

$$|\psi_s\rangle = \frac{1}{\sqrt{p(s)}} \sum_{t|q_{i_1},\ldots,q_{i_m}=s} \alpha_t |t\rangle.$$

Strong correlation between entangled qubits can be exposed by quantum measurement. Here we only consider the Bell state as an example. If a two qubit register is in the state $|\beta\rangle$ given by Eq. (4), and a measurement is performed on the first qubit, then we will obtain result 0 with probability 1/2, leaving the post-measurement state $|\beta_0\rangle = |00\rangle$, and 1 with probability 1/2, leaving the post-measurement state $|\beta_1\rangle = |11\rangle$. It is worth noting that after the measurement, the first and second qubits will always be in the same state.

2.4. The Deutsch–Jozsa algorithm

Quantum computation offers the possibility of considerable speedup over classical computation by exploring the power of superposition of quantum states. This can be illustrated very well by the Deutsch–Jozsa algorithm, which was designed in [32] to solve the following:

Deutsch's problem. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be constant if $f(x)$ equals 0 or 1 for all values of x. It is said to be balanced if $f(x)$ equals 0 for exactly half of all the possible x, and 1 for the other half. Suppose we know that a function f is either constant or balanced. The problem is how to determine with certainty whether f is constant or a balanced function.

A deterministic classical algorithm to solve the Deutsch's problem can be described as follows: (i) Select a value $x \in \{0,1\}^n$; (ii) Calculate $f(x)$; (iii) Repeat (i) and (ii). It is clear that at worst, the algorithm requires $2^{n-1}+1$ evaluations of f.

The Deutsch–Jozsa algorithm is an ingenious combination of quantum parallelism and interference. The presentation of this algorithm given here follows [74] and it can be described as follows:

- **Inputs:** A black box U_f which performs the transformation:

$$|x\rangle|y\rangle \to |x\rangle\big|y \oplus f(x)\big\rangle \quad \big(x \in \{0,1\}^n, y \in \{0,1\}\big).$$

- **Outputs:** 0 if and only f is constant.
- **Runtime:** One evaluation of U_f. Always succeeds.
- **Procedure:**

1. $|0\rangle^{\otimes n}|1\rangle$
2. $\xrightarrow{H^{\otimes(n+1)}} \dfrac{1}{\sqrt{2^n}} \displaystyle\sum_{x\in\{0,1\}^n} |x\rangle|-\rangle$
3. $\xrightarrow{U_f} \dfrac{1}{\sqrt{2^n}} \displaystyle\sum_x (-1)^{f(x)}|x\rangle|-\rangle$
4. $\xrightarrow{H^{\otimes n} \text{ on the first } n \text{ qubits}} \displaystyle\sum_z \frac{\sum_x (-1)^{x\cdot z + f(x)}}{2^n}|z\rangle|-\rangle$
5. $\xrightarrow{\text{measure on the first } n \text{ qubits in the computational basis}} z$

The algorithm uses a quantum register consisting of $n+1$ qubits. The first n qubits are initialized in the classical state $|0\rangle$ and the last qubit is initialized in the classical state $|1\rangle$. The main purpose of Step 2 is to produce an equal superposition of all computational basis states. This is realized by applying a Hadamard transformation, which is implemented by n Hadamard gates, to the first n qubits. In Step 3, the values of function f for all input $x \in \{0,1\}^n$ are evaluated simultaneously by using the quantum gate U_f once. This is quantum parallelism! It is worth noting that in the definition of U_f, the value $f(x)$ occurs in $|y \oplus f(x)\rangle$, but in this step it is moved to the exponent in $(-1)^{f(x)}$ by cleverly putting the last qubit in the state $|-\rangle$ and observing that $|0 \oplus u\rangle - |1 \oplus u\rangle = (-1)^u(|0\rangle - |1\rangle)$. The aim of this trick will become clear in the next step. To get a computational outcome, we have to do a measurement. If we directly measure the first n qubits in the computational basis at this stage, only $f(x)$ for a single value of x can be obtained, and the power of quantum parallelism vanishes. Fortunately, quantum interference provides the ability to extract information about more than one value of $f(x)$ from a superposition state. To understand quantum interference, we consider a general superposition $\sum_x \alpha_x|x\rangle$. If we directly measure it, we can only get local information about α_x for a single value of x. However, if we first perform a suitably chosen unitary transformation:

$$U|x\rangle = \sum_z u_{xz}|z\rangle \quad \text{for all } x$$

on it, then

$$U\left(\sum_x \alpha_x|x\rangle\right) = \sum_x \alpha_x\left(\sum_z u_{xz}|z\rangle\right) = \sum_z \left(\sum_x \alpha_x u_{xz}\right)|z\rangle.$$

Now measuring $U(\sum_x \alpha_x|x\rangle)$ we can obtain certain global information about all α_x's through amplitude $\sum_x \alpha_x\beta_{xz}$ for a single value of z. You can see that Step 4 is exactly an application of quantum interference. Note that moving $f(x)$ to the exponent in Step 3 allows us to conveniently combine it with the amplitudes produced by the Hadamard transformation in Step 4. Finally, we observe that the amplitude of $|0\rangle^{\otimes n}|-\rangle$ is

$$\frac{1}{2^n}\sum_x (-1)^{f(x)},$$

which equals 0 when f is balanced and ± 1 when f is constant.

3. A survey of quantum computation

This section is definitely not a balanced survey, and the emphasis will be given to those areas that I am familiar with although they may not be the most active ones. Of course, physical implementations of scalable and functional quantum computers is one of the most important problems in quantum computation. But this topic will not be touched on in this paper simply because it lies outside my expertise. Another important topic not considered in this section for the same reason is quantum error-correction and fault-tolerant quantum computation. For an excellent exposition of these topics, see [74], Chapters 7 and 10.

At this moment, most of the topics reviewed in this section have no obvious links to AI, but I hope the reader will find some interesting connections between them and AI.

3.1. Models of quantum computation

3.1.1. Quantum Turing machine and quantum automata

The models of quantum computation have their ancestors from the studies of connections between physics and computation. In 1973, to understand the thermodynamics of classical computation Bennet [13] noted that a logically reversible operation does not need to dissipate any energy and found that a logically reversible Turing machine is a theoretical possibility. In 1980, Benioff [11] constructed a quantum mechanical model of a Turing machine. His construction is the first quantum mechanical description of computer, but it is not a real quantum computer because the machine may exist in an intrinsically quantum state between computation steps, but at the end of each computation step the tape of the machine always goes back to one of its classical states. The first truly quantum Turing machine was described by Deutsch [30] in 1985. In his machine, the tape is able to exist in quantum states too. This is different from Benioff's machine. A thorough exposition of the quantum Turing machine is given in [14].

In the realm of classical computation, finite automata and pushdown automata have been widely applied in the design and implementation of programming languages. Several quantum generalizations of finite and pushdown automata were introduced by Kondas and Watrous [63], Gudder [54], and Moore and Crutchfield [69] in the late 1990's. Their definitions of quantum automata differ mainly in where quantum measurements are allowed. For example, a quantum automaton introduced in [69] may be observed only after all input symbols have been read, whereas a quantum automaton in [63] is allowed to be observed after reading each symbol. The most general model of quantum finite automata was proposed independently by Bertoni, Mereghetti and Palano [15] and Ciamarra [25], and it admits any sequence of unitary transformations and measurements.

Recently, some applications of quantum automata have been found; for example, Nishimura and Yamakami [76] provided a direct application of quantum automata to interactive proof systems. But it seems not the case that quantum automata can be used in compiling of quantum programming languages.

3.1.2. Quantum circuits

The circuit model of quantum computation was also proposed by Deutsch [31]. Roughly speaking, a quantum circuit consists of a sequence of quantum gates connected by quantum wires that carry qubits. Yao [102] showed that quantum circuit model is equivalent to a quantum Turing machine in the sense that they can simulate each other in polynomial time. Since then, quantum circuits has become the most popular model of quantum computation in which most of the existing quantum algorithms are expressed.

Synthesis of quantum circuits is crucial for quantum computation due to the fact that in current technologies it is very difficult to implement quantum gates acting on three or more qubits. As early as in 1995, it was shown that any quantum gate can be (approximately) decomposed to a circuit consisting only of the CNOT gates and a small set of single qubit gates [10]. Recently, some more efficient synthesis algorithms for quantum circuits have been found; see for example [90].

Some authors initiated the studies of simplification and optimization of quantum circuits. The aim is to develop methods and techniques to reduce the number of quantum gates in a quantum circuit and the depth of a quantum circuit. Due to the difficulty of implementing large quantum circuits, this problem is even more important in quantum computation than in classical computation. The current research includes: (1) ad hoc techniques for simplifying quantum circuits for some special classes of computations; for example, Meter and Itoh [67] proposed a compaction method for quantum circuits of modular exponentiation; (2) general techniques; for example, Maslov et al. [66] introduced a local optimization technique for quantum circuits based on templates.

In the current literature, quantum circuits are mainly drawn as circuit graphs, and reasoning about quantum circuits is usually carried out by thorough inspection of their actions on various input states. It is obvious that the circuit graphs for complicated quantum algorithms would be too big to be drawn. To provide the facility of doing algebraic manipulation on quantum circuits, an algebraic language was designed [109] in which quantum circuits can be conveniently expressed in a way similar to that of representing classical circuits by Boolean expressions. However, an algebraic language is not enough to support algebraic manipulation on and reasoning about quantum circuits. We still need to establish various algebraic laws for quantum circuits that will play a role similar to switching algebra or more generally Boolean algebra for classical circuits. A preliminary attempt toward a comprehensive algebra of quantum circuits was made in [110].

3.1.3. Adiabatic quantum computation

Quantum Turing machine, quantum automata and quantum circuits are quantum generalizations of their classical counterparts. Recently, several novel models of quantum computation have been conceived and they have no evident classical analogues, one of such models is adiabatic quantum computation proposed by Farhi, Goldstone, Gutmann and Sipser [41]. Different from all of the other models considered in this section, which are discrete-time models, adiabatic quantum computation is a continuous-time model of computation. It is based on the adiabatic theorem in quantum physics. In adiabatic quantum computation, the evolution of the quantum register is governed by a Hamiltonian that varies slowly. The state of the system is prepared at the beginning in the ground state of the initial Hamiltonian. The solution of a computational problem is then encoded in the ground state of the final Hamiltonian. The quantum adiabatic theorem guarantees that the final state of the system will differ from the ground state of the final Hamiltonian by a negligible amount provided the Hamiltonian of the system evolves slowly enough. Thus the solution can be obtained with a high probability by measuring the final state. The adiabatic model provides a new way of designing quantum algorithms; for example, the Grover's algorithm has been recast in the adiabatic model.

3.1.4. Measurement-based quantum computation

Another model of quantum computation without a classical counterpart is measurement-based computation. In the quantum Turing machine and quantum circuits, measurements are mainly used at the end to extract computational outcomes from quantum states. However, Raussendorf and Briegel [83] proposed a one-way quantum computer and Nielsen [73] and Leung [65] introduced teleportation quantum computation, both of them suggests that quantum measurements can play a much more important role in quantum computation. In a one-way quantum computer, universal computation can be realized by one-qubit measurements together with a special entangled state, called a cluster state, of a large number of qubits. Teleportation quantum computation is based on Gottesman and Chuang's idea of teleporting quantum gates [51] and allows us to realize universal quantum computation using only projective measurement, quantum memory, and preparation of the $|0\rangle$ state. The measurement-based model offers new possibilities for the physical implementation of quantum computation. Recently, Danos, Kashefi and Panangaden [28] proposed a calculus for formally reasoning about (programs in) measurement-based quantum computation.

3.1.5. Topological quantum computation

A crucial challenge in constructing large quantum computers is quantum decoherence. In 1997, topological quantum computation was proposed by Kitaev [61] as a model of quantum computation in which a revolutionary strategy is adopted to build significantly more stable quantum computers. This model employs two-dimensional quasiparticles, called anyons,

whose world lines forms braids, which are used to construct logic gates of quantum computers. The key point is that small perturbations do not change the topological properties of these braids. This makes quantum decoherence simply irrelevant for topological quantum computers. For an excellent exposition of topological quantum computation, see [82].

3.1.6. Distributed quantum computation

The earliest suggestions for distributed quantum computation can be traced back to Grover [53] and Cleve and Buhrman [26] among others. One of the major motivations arises from the extreme difficulty of the physical implementation of functional quantum computers. A natural idea is to use the physical resources of two or more small capacity quantum computers to simulate a large capacity quantum computer; for example, a distributed implementation of Shor's quantum factoring algorithm is presented in [103]. Another major motivation comes from the studies of quantum communication. By employing quantum mechanical principles, some provably secure communication protocols have been proposed, and quantum communication systems using these protocols are already commercially available. To provide formal techniques for verifying quantum communication protocols, Gay and Nagarajan [49] defined a language CQP (Communicating Quantum Processes) and Jorrand and Lalire [60] defined a language QPAlg (Quantum Process Algebra) which are obtained from the pi-calculus and a classical process algebra similar to CCS, respectively, by adding primitives for quantum gates and measurements and allowing transmission of qubits. More recently, bisimulation semantics for quantum process algebras were introduced in [44,112]. In particular, a notion of approximate bisimulation is proposed to provide a formal tool for describing robustness of quantum processes against inaccuracy in the implementation of its elementary gates. The third major motivation is to find quantum algorithms for solving paradigmatic problems from classical distributed computation. For example, it is well known that no classical algorithms can exactly solve the leader election problem in anonymous networks, but Tani, Kobayashi and Matsumoto [96] and D'Hondt and Panangaden [38] developed a quantum algorithm that can solve it for any network topology in polynomial communication/time complexity provided certain entanglement exists between the involved parties.

3.2. Logical foundations of quantum computation

3.2.1. Categorical quantum logic

Currently, quantum algorithms and communication protocols are expressed mainly at the very low level of quantum circuits. We learned in classical computation that high-level description is very useful for design and analysis of algorithms and protocols because it enables us to think about a problem that we intend to solve in a conceptual way, rather than the details of implementation. However, high-level description techniques are still lacking in quantum computation [1]. As a response to the requirement of high-level description in quantum information science, Abramsky and Coecke [5] proposed a category-theoretic axiomatization of quantum mechanics by employing formal tools mainly developed in computer science, especially Abramsky's previous work on semantics of concurrency and geometry of interaction. More concretely, the standard von Neumann's Hilbert space formalism of quantum mechanics can be recast in the abstract language of strongly compact closed categories with biproducts. What is particularly interesting is that a categorical approach to quantum theory provides effective methods for high-level description and verification of quantum communication protocols, including teleportation, logic-gate teleportation, and entanglement swapping [2]. In particular, it provides a new insight that nonlocal classical communication can be elegantly depicted in distributivity. Furthermore, a logic of strongly compact closed categories with biproducts in the form of proof-net calculus is developed by Abramsky and Duncan [3] as a categorical quantum logic. It is suitable for high-level reasoning about quantum processes. More recently, Heunen and Jacobs [55] investigated quantum logic from the perspective of categorical logic, and they showed that kernel subobjects in dagger kernel categories precisely capture orthomodular structure.

3.2.2. Quantum lambda calculus

The lambda calculus is a formalism of high-order functions and it is a logical basis of some important classical functional programming languages such as LISP, Scheme, ML and Haskell. A quantum generalization of λ-calculus was first introduced by Tonder [97]. The no-cloning property of quantum data makes quantum lambda calculus closely related to linear lambda calculus developed by the linear logic community. In a series of papers [88], Selinger and Valiron systematically develop quantum lambda calculus. In particular, quantum lambda calculus was used by them [89] to provide a fully abstract model for the linear fragment of a quantum functional programming language, which is obtained by adding higher-order functions into Selinger's quantum flowchart language QFC [87].

3.2.3. Quantum computational logic

Quantum logic was proposed by Birkhoff and von Neumann [17] as a logic of quantum mechanics about 70 years ago. Propositions in quantum logic are interpreted as closed subspaces of the state space (a Hilbert space) of a quantum system, or their algebraic abstraction, elements of an orthomodular lattice, and logical connectives are then naturally interpreted as the operations in the orthomodular lattice. The basic idea of this semantics of quantum logic stemmed from von Neumann's projective measurement theory. Inspired by the rapid development of quantum computation, Cattaneo, Dalla Chiara, Giuntini and Leporini [21] introduced a quantum computational logic in which propositions are interpreted as states of quantum registers and logical connectives are interpreted as quantum gates or operations that can be conveniently expressed in terms

of quantum gates. This logic can be used to describe and reason about quantum circuits. It seems that some interesting connection between quantum computational logic and the work on algebra of quantum circuits [109,110] exists and worths some further studies.

3.2.4. Theory of computation based on quantum logic

The (meta)logic underlying classical theory of computation is Boolean (two-valued) logic. Birkhoff and von Neumann's quantum logic [17] is understood as a logic whose truth values are taken from an orthomodular lattice. The major difference between Boolean logic and quantum logic is that the latter does not enjoy distributivity in general. Automata theory based on quantum logic was developed in [104,105]. Various properties of automata are carefully reexamined in the framework of quantum logic by employing an approach of semantic analysis. It is found that universal validity of many important properties of automata depends heavily upon distributivity of the underlying logic. This indicates that these properties do not universally hold in the realm of quantum logic. On the other hand, we show that a local validity of them can be recovered by imposing a certain commutativity to the (atomic) statements about the automata under consideration. This reveals an essential difference between classical automata theory and automata theory based on quantum logic.

Automata theory based on quantum logic can be seen as a logical abstraction of quantum automata discussed in Section 3.1.1. Indeed, the relation between quantum automata and automata theory based on quantum logic is quite similar to that between von Neumann's Hilbert space formalism of quantum mechanics and quantum logic.

3.3. Quantum algorithms

Research on quantum algorithms has been the driving force of the whole field of quantum computation because some quantum algorithms indicate that quantum computation may provide considerable speedup over classical computation. Unfortunately, I am not an expert in quantum algorithms and thus can only give a very brief survey of this area. Three classes of quantum algorithms have been discovered, which show an advantage over known classical algorithms: (1) algorithms based on quantum Fourier transforms, e.g. the Deutsch–Jozsa algorithm and Shor's algorithm for factoring and discrete logarithm; (2) quantum search algorithms, that is, Grover's algorithms and its extensions; (3) quantum algorithms for simulation of quantum systems, with the basic idea tracing back to Feynman [47]. For elaborations of these algorithms, see [74], Chapters 5 and 6 and Section 4.7. It is quite disappointing that no new classes of quantum algorithms have been proposed for 15 years. Shor [92] gave some explanations for why so few quantum algorithms surpassing their classical counterparts have been found and pointed out several lines of research that might lead to discovery of new quantum algorithms.

3.4. Quantum computer architectures

Progress in the techniques of quantum devices has made people widely believe that large-scalable and functional quantum computers will eventually be built. Architecture design will become more and more important as the size of quantum computers grows. Quantum computer architecture is another area that I am not familiar with. What I know is merely that research in quantum computer architectures is still in its infancy and there are only few papers devoted to this topic. Copsey et al. [27] proposed a scalable, silicon based architecture of quantum computer. A related work is that Svore et al. [94] introduced a layered software architecture for quantum computer design tools.

3.5. Quantum programming

Our experiences with classical computation suggest that when quantum computers become available in the future, quantum softwares will play a key role in exploiting their power. Unfortunately, today's software development methodologies and techniques are not suited to quantum computers due to essential differences between the nature of the classical world and that of the quantum world. To lay a solid foundation for tomorrow's quantum software development techniques, it is critically essential to pursue systematic research into quantum programming [48,70,80].

The earliest proposal for a quantum programming language was made by Knill [62]. The first real quantum programming language, QCL, was proposed by Ömer [77]; he also implemented a simulator for this language. A quantum programming language in the style of Dijkstra's guarded-command language, qGCL, was designed by Sanders and Zuliani [84]. A quantum extension of C++ was proposed by Bettelli et al. [16], and implemented in the form of a C++ library. The first quantum language of the functional programming paradigm, QFC, was defined by Selinger [87] based on the idea of classical control and quantum data. A quantum functional programming language with quantum control was introduced in [7].

Understanding behaviors of complex quantum program constructs is crucial for quantum programming. Some high-level control features such as loop and recursion are provided in Selinger's language QFC [87]. In [111], a general scheme of quantum loop programs was introduced. The essential difference between quantum loops and classical loops comes from quantum measurements in the loop guards. In a fixed finite-dimensional state space, a necessary and sufficient condition under which a quantum loop program terminates on a given input was found by employing Jordan normal form of complex matrices. In particular, it was proved that a small disturbance either on the unitary transformation in the loop body or on the measurement in the loop guard can make any quantum loop (almost) terminate, provided that some obvious dimension restriction is satisfied.

The fact that human intuition is much better adapted to the classical world than the quantum world suggests that programmers may commit more faults in designing programs for quantum computers than programming classical computers. Thus, it seems that giving clear and formal semantics to quantum programming languages and providing formal methods for reasoning about quantum programs are even more critical than in classical computation. Since it provides a goal-directed program development strategy, predicate transformer semantics has a wide influence in classical programming methodology. Two approaches to predicate transformer semantics of quantum programs have been proposed in the literature. The first was proposed by Sanders and Zuliani [84] in designing qGCL, where quantum computation is reduced to probabilistic computation by the observation (measurement) procedure. Thus, predicate transformer semantics developed for probabilistic programs can be conveniently applied to quantum programs. The second was proposed by D'Hondt and Panangaden in [37], where the notion of predicate is directly taken from quantum mechanics; that is, a quantum predicate is defined to be an observable (a Hermitian operator) with eigenvalues within the unit interval. The forward operational semantics of quantum programs are described by super-operators (completely positive operators), and a beautiful duality between state-transformer (forward) and predicate-transformer (backward) semantics is then achieved by employing the Kraus representation theorem for super-operators. One of the advantages of the second approach is that it provides a very natural framework to model and reason about quantum programs. It seems that a link between these two approaches to quantum predicate transformer semantics can be established through the Gleason theorem [50].

It should be emphasized that the subject of quantum programming methodology is not a simple and straightforward generalization of its classical counterpart. Some completely new phenomena arise in the quantum case. These problems stem from the "weird" nature of quantum systems. For example, no-cloning of quantum data means that the typing systems of quantum programming languages are essentially different from those of classical computation [49]. It was observed in [107] that noncommutativity is a major obstacle in developing D'Hondt and Panangaden's predicate transformer semantics because various logical operations of quantum weakest preconditions will be needed in reasoning about complicated quantum programs, but defining these operations requires commutativity between the quantum predicates involved [99]. It was suggested in [108] to focus attention on a special class of quantum predicates, namely projection operators. This allows us to use rich mathematical methods developed in Birkhoff–von Neumann quantum logic [17] and Takeuti's quantum set theory [95]. In particular, the Takeuti's notion of commutator helps us to establish various healthiness conditions of quantum programs, e.g. termination law and conjunctivity.

Some proof systems for reasoning about quantum programs have been proposed. Baltag and Smets [9] presented a dynamic logic formalism of information flows in quantum systems. Brunet and Jorrand [19] introduced a way of applying Birkhoff and von Neumann's quantum logic [17] to the study of quantum programs by expanding the usual propositional languages with new primitives representing unitary transformations and quantum measurements. In [22], Chadha, Mateus and Sernadas proposed a Hoare-style proof system for reasoning about imperative quantum programs using a quantitative state logic, but only bounded iterations are allowed in their programming language. Some useful proof rules were proposed in [43] for purely quantum programs within a finite-dimensional state space. Furthermore, a full-fledged Hoare logic for both partial and total correctness of quantum programs was developed in [106].

The existing programming languages for quantum computation are designed and their semantics is investigated according to ordinary circuit models of quantum computation, except the measurement calculus [28] was introduced for reasoning about programs in measurement-based model. It seems that the principles and semantics of programming languages for adiabatic and topological quantum computers will essentially differ from those for the circuit model. This area of research is essentially green field, and much exciting work is yet to be done.

4. Potential applications of quantum computation in AI

Of course, it will be very exciting for both quantum computation researchers and AI researchers to use quantum computation in AI. Quantum computation researchers hope to find more quantum algorithms demonstrating significant speedup over classical algorithms. They are looking for new problems suited to this purpose, and some AI problems seems to be good candidates. On the other hand, the AI community believes that quantum computation shows significant potential for solutions to currently intractable problems. Indeed, 10 years ago, the "Trends and Controversies" of the July/August issue of the magazine *IEEE Intelligent Systems* was devoted to the possibility of combining quantum computation and AI [56]. Also, some quantum computation researchers were invited to present Tutorials at *IJCAI* conferences. To the best of my knowledge, however, not much progress has been made in this direction up to now. Perhaps, this is because not much effort has been expended, the majority of AI community may think that quantum computing technology is still in its infancy, and it is too early to consider how quantum computation can be used in AI. So, what we can do in this section is to explore some of possibilities of applying quantum computation in AI rather than to review the existing applications of quantum computation in AI.

4.1. Quantum algorithms for learning

Maybe the only area where quantum computation and AI have already met in a fruitful way is machine learning. There are several papers devoted to quantum generalization of computational learning theory. Their aim is to find some quantum algorithms that are more efficient than the existing classical algorithms for learning of certain classical objects, such as

Boolean functions. This research is closely related to quantum complexity theory [14]. I am not an expert in this area, but fortunately a good survey of it already exists [18]. This survey is not new, but it is quite comprehensive.

A dual topic is learning objects in the quantum world using mainly classical methods (together with quantum measurements), and it will be discussed in Section 5.5.

4.2. Quantum algorithms for decision problems

Many decision problems can be formulated in terms of decision trees. Farhi and Gutmann [42] showed that quantum algorithms based on Hamiltonian evolution can solve the decision problems represented by a class of decision trees exponentially faster than classical random walks. But this does not imply any advantage of quantum computation over classical computation for this class of problems because they can also be solved very quickly by other classical algorithms.

4.3. Quantum search

Much of the early AI research was concerned with search techniques. This may be because on the one hand, many AI problems can be reduced to searching; for example, planning, scheduling, theorem proving and information retrieval, and on the other hand, computers can do these kinds of tasks much faster than humans. The Grover algorithm [52] shows that quantum computers can do it even faster than classical computers. Naturally, people expect that quantum computation will be widely used in AI to solve various search-related problems. It is believed that quantum searching will be one of the first quantum computing techniques that play an important role in AI. In 1999, Hogg [57] discussed the problem of how quantum search algorithms can be applied in AI in detail. But up to now, 10 years later, few successful applications of quantum searching in AI have been reported.

4.4. Quantum game theory

Game theory is being used in AI progressively more and more, especially in multi-agent systems and distributed AI. Recently, quantum extensions of game theory have been proposed in a series of papers; for example, Eisert, Wilkens and Lewenstein [39] introduced quantization of nonzero sum games with two players, and Benjamin and Hayden [12] introduced quantum games with more than two players. Miakisza, Piotrowski and Sładkowskic [68] argued that quantum game theory [39] offers new tools for solutions of some problems in AI.

Other possibilities of applying quantum computation in AI include:

- Representing knowledge in the way of quantum superposition, and speeding up knowledge reasoning by quantum parallelism.
- Using quantum communication and distributed quantum computation in multi-agent systems; in particular, using entanglement for coordination.

5. Interplay between quantum theory and AI

Research arising from the interplay between quantum theory and AI can be roughly classified into two categories: (1) Using some ideas from quantum theory to solve certain problems in AI; and (2) Conversely, applying some ideas developed in AI to quantum theory. We first see how ideas from quantum theory be used in AI by considering two typical examples.

5.1. Semantic analysis

Some similarities between the mathematical structure used by the AI community in semantic analysis of natural language and those employed in quantum mechanics were observed in [5]. But these similarities exposed in [5] seems very superficial, and they do not convince me to believe that a certain intrinsic connection exists between semantic analysis and quantum mechanics because it is not surprising that the same mathematical tools can be applied in unrelated domains, and indeed universal effectiveness is exactly one of the most important advantages of mathematics. On the other hand, however, observation of these similarities is still useful since by analogy it may provide hints as to how one can borrow some ideas from the well-established subject of quantum mechanics in semantic analysis or even more broadly in AI. Furthermore, if some semantic aspects of natural languages can be properly expressed in the framework of quantum theory, e.g. ambiguity by superposition, then the fact that quantum algorithms are especially suited to simulation of quantum systems suggests that quantum computation might considerably speedup natural language processing.

5.2. Entanglement of words in natural languages

Nelson, McEvoy and Pointer [72] noticed that word associations in natural languages can display 'spooky action at a distance behavior'. Bruza et al. [20] proposed a model of word associations in terms of tensor products so that 'spooky activation at a distance' can be described in a way similar to quantum entanglement.

We now turn to consider the inverse problem: how can some ideas developed in AI be used in quantum theory. The research on this problem can also be seen from another point of view. The current AI community is mainly devoted to develop computing techniques that implement intelligence for dealing with problems in the classical world. The research considered in the following subsections can be thought of as AI techniques that implement intelligence for coping with problems in the quantum world. In fact, the quantum counterparts of some basic AI problems such as learning and pattern recognition have been identified and intensively studied by physicists working in the fields of quantum information. It seems that AI researchers do not know much about this kind of work. I believe that AI researchers' participation in understanding quantum information will accelerate the development of this area, and the methodologies and techniques developed by AI researchers will help quantum physicists.

5.3. Quantum Bayesian networks

Statistical inference is at the heart of quantum theory due to the essential probabilistic nature of quantum systems. Bayesian methods have been widely used in statistical inference in the classical world. Recently, several versions of quantum Bayes rule have been derived in the physics literature; see for example [86].

Bayesian networks are graph models for representing and reasoning about probability information and widely used in AI. It is hoped that this kind of graph model can be adopted in reasoning about the behaviors of large systems in the quantum world. Tucci [98] introduced a quantum generalization of Bayesian networks in which complex amplitudes rather than (conditional) probabilities are assigned to its nodes and used it to calculate probabilities for some physical experiments.

Pearl [78] introduced the notion of causal Bayesian networks which augments Bayesian networks with a set of local operations that specify how probability distributions behave with respect to external interventions. To provide a graph model of causality in the quantum world, Laskey [64] defined a notion of quantum causal networks where the local operations are represented by super-operators that are a popular mathematical formalism of the dynamics of open quantum systems.

5.4. Recognition and discrimination of quantum states and quantum operations

Pattern recognition is an important area of AI, and discrimination of objects can be seen as a special case of pattern recognition. However, only recognition and discrimination of classical objects have been considered by AI researchers. In the last 20 years, a large amount of work on discrimination and recognition of quantum states and quantum operations has been conducted by physicists without knowing much about existing AI work.

Unambiguous discrimination of quantum states may be formulated as follows: a system is prepared in a number of known, finite set of pure quantum states $|\varphi_1\rangle, \ldots, |\varphi_n\rangle$, and we hope to determine what quantum state the system is actually in with the requirement that once a result is reported, it must be true. This problem was first considered by Ivanovic [58] for the case of $n = 2$. The general case was examined by Chefles [23]. It was shown in [93] that the optimal success probability of discrimination is mathematically equivalent to the well-known semidefinite programming problem. An estimation of success probability was given in [46,113]. The problem of discrimination of quantum states was generalized in [45] to the case of mixed states.

Recently, discrimination of quantum operations has received considerable attention. The problem of discriminating (global) unitary transformations (quantum gates) was solved by Acín [4] and D'Ariano, Presti and Paris [29], and studies on discrimination of quantum measurements were initiated in [59]. The general case of (global) quantum operations represented by super-operators was considered in [101]. In particular, a complete characterization of perfect distinguishability of quantum operations was achieved in [36] by discovering a feasible necessary and sufficient condition under which an unknown quantum operation secretly chosen from a finite set of quantum operations can be identified perfectly and by designing an optimal protocol for such a discrimination with a minimal number of queries. A particularly interesting problem is discrimination of quantum operations acting on a multipartite quantum system by local operations and classical communication (LOCC for short). Surprisingly, it is proved in [33–35] that entanglement is unnecessary for this kind of discrimination of unitary operators although it had been believed that entanglement was necessary.

The pattern recognition problem for quantum states was considered by Sasaki and Carlini [85]: Given a set of template quantum states $|\varphi_1\rangle, \ldots, |\varphi_n\rangle$. Decide which of them is closest to an input state $|\psi\rangle$. An essential difference between quantum and classical pattern recognition is that in the quantum case multiple copies of the template and input states may be required since quantum measurements are employed in the recognition strategy and they usually change the states of the measured systems. A Bayesian learning method was proposed in [85] to accomplish the task of quantum pattern recognition.

5.5. Learning of quantum states and quantum operations

The problem dealt with in this section is different from that considered in Section 4.2 where classical objects are learned but the learning algorithms are quantum. Here, the learned objects are quantum [6]. To give the reader a taste, we consider a simple example of supervised concept learning. In the classical case, the training data set is usually given in the form of $D = \{(x_i, c(x_i)) : \ i = 1, \ldots, n\}$, where x_i's are instances, and $c(x_i) = 1$ or 0 for all i ($c(x_i) = 1$ means that x_i is a positive example and $c(x_i) = 0$ means that x_i is a negative example). In the quantum case, the instances x_i's are replaced by quantum states, say $|\psi_i\rangle$'s. If the descriptions of quantum instances $|\psi_i\rangle$'s are given classically, then the quantum learning problem

immediately degenerates to a classical learning problem. More interesting is the case where no classical descriptions of these quantum states are available. To learn a concept from the quantum training set, one needs to extract classical information from them and then certain quantum measurements have to be performed on these quantum states. Since these quantum measurements will destroy the original quantum states, multiple copies of these quantum states may be required. This is contrary to the classical case.

Quantum state tomography [100] can be seen as a kind of quantum learning. The scenario is as follows: There is a physical process that can produces a quantum state repeatedly. We prepare as many copies of the state as needed by applying this process. Our goal is to learn a description of the state from the measurement outcomes performed on these copies. A similar problem for quantum operations is known as quantum process tomography of which a theory was developed by Chuang and Nielsen [24] and Poyatos, Cirac and Zoller [81].

The studies of learning in the quantum world are still at the initial stage. Quantum generalizations of various sophisticated machine learning methods are entirely untouched. This presents a good opportunity to AI researchers because physicists may not be aware of these methods.

Other research arising from the interplay between quantum theory and AI include:

- Quantum neural networks, see for example [40].
- Quantum genetic algorithms, see for example [71].

There are many interesting topics for which a proper problem statement and an appropriate setting are still unknown. Here I only mention:

- Spatial reasoning in the quantum world.
- Constraint satisfaction of quantum states.

Certain interplay between quantum theory and AI has been examined in this section, but a much deeper connection between these two subjects may come from macroscopic quantum effects in the brain as is explored by Penrose [79]. But a serious consideration of this issue is outside the author's expertise.

6. Conclusion

This paper identifies three classes of opportunities for AI researchers at the intersection of quantum computation, quantum theory and AI:

- Design quantum algorithms to solve problems in AI more efficiently;
- Develop more effective methods for formalizing problems in AI by borrowing ideas from quantum theory;
- Develop new AI techniques to deal with problems in the quantum world.

The first class of research is still in the initial stage of development, and not much progress has been made. Shor [92] listed some reasons to explain why quantum algorithms are so hard to discover. Unfortunately, these reasons are valid for the problems in AI too. Some fragmented and disconnected research belonging to the second class have a long history, and some basic ideas can even be traced back to Niels Bohr. In recent years, research in this class has become very active, especially through the *International Symposium on Quantum Interaction* (2007–2009). But it seems that some of these works are quite superficial, and deeper theoretical analysis of the formal methods developed in these works are needed. In particular, more experimental research is required to test the effectiveness. It appears that research in the third class is making steady progress. My main concern is whether the AI techniques developed in this class of research will be useful in quantum physics and will be appreciated by physicists. Certainly, collaboration between AI researchers and physicists will highly benefit the development of this area. Perhaps, experience from bioinformatics can be used for reference where close collaboration between computer scientists and biologists frequently happens and leads to high impact research.

Acknowledgements

The author is deeply indebted to Professors Mary-Anne Williams and Vaughan R. Pratt for offering invaluable comments and suggestions. He is very grateful to Dr. Sanjiang Li, Dr. Yuan Feng and Dr. Runyao Duan for their stimulating discussions and invaluable suggestions. The author would like to express his sincere thanks to Professor Chengqi Zhang, Director of the Center of Quantum Computation and Intelligent Systems, University of Technology, Sydney, for providing the excellent working environment.

References

[1] S. Abramsky, High-level methods for quantum computation and information, in: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, pp. 410–414.
[2] S. Abramsky, B. Coecke, A categorical semantics of quantum protocols, in: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, pp. 415–425.

[3] S. Abramsky, R. Duncan, A categorical quantum logic, Mathematical Structures in Computer Science 16 (2006) 469–489.
[4] A. Acín, Statistical distinguishability between unitary operations, Physical Review Letters 87 (2001), art. no. 177901.
[5] D. Aerts, M. Czachor, Quantum aspects of semantic analysis and symbolic artificial intelligence, Journal of Physics A: Mathematical and General 37 (2004) L123–L132.
[6] E. Aïmeur, G. Brassard, S. Gambs, Machine learning in a quantum world, in: L. Lamontagne, M. Marchand (Eds.), Proceedings of Canadian AI 2006, in: LNAI, vol. 4013, Springer, 2006, pp. 431–442.
[7] T. Altenkirch, J. Grattage, A functional quantum programming language, in: Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS), 2005, pp. 249–258.
[8] S.L. Andresen, John McCarthy: Father of AI, IEEE Intelligent Systems (September/October 2002) 84–85.
[9] A. Baltag, S. Smets, LQP: The dynamic logic of quantum information, Mathematical Structures in Computer Science 16 (2006) 491–525.
[10] A. Barenco, C. Bennet, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, H. Weinfurter, Elementary gates for quantum computation, Physical Review A 52 (1995) 3457–3467.
[11] P.A. Benioff, The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, Journal of Statistical Physics 22 (1980) 563–591.
[12] S.C. Benjamin, P.M. Hayden, Multiplayer quantum games, Physical Review A 64 (2001), art. no. 030301.
[13] C.H. Bennet, Logical reversibility of computation, IBM Journal of Research and Development 17 (1973) 525–532.
[14] E. Bernstein, U. Vazirani, Quantum complexity theory, SIAM Journal on Computing 26 (1997) 1411–1473.
[15] A. Bertoni, C. Mereghetti, B. Palano, Quantum computing: 1-way quantum automata, in: Z. Esik, Z. Fulop (Eds.), Proceedings of 7th International Conference on Developments in Language Theory, in: LNCS, vol. 2710, Springer, 2003, pp. 1–20.
[16] S. Bettelli, T. Calarco, L. Serafini, Toward an architecture for quantum programming, The European Physical Journal D 25 (2003) 181–200.
[17] G. Birkhoff, J. von Neumann, The logic of quantum mechanics, Annals of Mathematics 37 (1936) 823–843.
[18] R. Bonner, R. Freivalds, A survey of quantum learning, in: R. Bonner, R. Freivalds (Eds.), Proceedings of the 3rd Workshop on Quantum Computation and Learning, 2002, pp. 106–119.
[19] O. Brunet, P. Jorrand, Dynamic quantum logic for quantum programs, International Journal of Quantum Information 2 (2004) 45–54.
[20] P. Bruza, K. Kitto, D. Nelson, C. McEvoy, Extracting spooky-activation-at-a-distance from considerations of entanglement, in: P. Bruza, et al. (Eds.), Proceedings of Third International Symposium on Quantum Interaction, in: LNCS, vol. 5494, Springer-Verlag, 2009, pp. 71–83.
[21] G. Cattaneo, M.L. Dalla Chiara, R. Giuntini, R. Leporini, An unsharp logic from quantum computation, International Journal of Theoretical Physics 43 (2004) 1803–1817.
[22] R. Chadha, P. Mateus, A. Sernadas, Reasoning about imperative quantum programs, Electronic Notes in Theoretical Computer Science 158 (2006) 19–39.
[23] A. Chefles, Unambiguous discrimination between linearly independent quantum states, Physical Letters A 239 (1998) 339–347.
[24] I.L. Chuang, M.A. Nielsen, Prescription for experimental determination of the dynamics of a quantum black box, Journal of Modern Optics 44 (1997) 2455–2467.
[25] M.P. Ciamarra, Quantum reversibility and a new model of quantum automaton, in: R. Freivalds (Ed.), Proceedings of the 13th International Symposium Fundamentals of Computation Theory, in: LNCS, vol. 2138, Springer, 2001, pp. 376–379.
[26] R. Cleve, H. Buhrman, Substituting quantum entanglement for communication, Physical Review A 56 (1997) 1201–1204.
[27] D. Copsey, M. Oskin, F. Impens, T. Metodiev, A. Cross, F.T. Chong, I.L. Chuang, J. Kubiatowicz, Toward a scalable, silicon-based quantum computing architecture, IEEE Journal of Selected Topics in Quantum Electronics 9 (2003) 1552–1569.
[28] V. Danos, E. Kashefi, P. Panangaden, The measurement calculus, Journal of the ACM 54 (2007), art. no. 8.
[29] G.M. D'Ariano, P.L. Presti, M.G.A. Paris, Using entanglement improves the precision of quantum measurements, Physical Review Letters 87 (2001), art. no. 270404.
[30] D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, Proceedings of The Royal Society of London A 400 (1985) 97–117.
[31] D. Deutsch, Quantum computational networks, Proceedings of The Royal Society of London A 425 (1989) 73–90.
[32] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation, Proceedings of The Royal Society of London A 439 (1992) 553.
[33] R.Y. Duan, Y. Feng, M.S. Ying, Entanglement is not necessary for perfect discrimination between unitary operations, Physical Review Letters 98 (2007), art. no. 100503.
[34] R.Y. Duan, Y. Feng, M.S. Ying, Local distinguishability of multipartite unitary operations, Physical Review Letters 100 (2008), art. no. 020503.
[35] R.Y. Duan, Y. Feng, Z.F. Ji, M.S. Ying, Distinguishing arbitrary multipartite basis unambiguously using local operations and classical communication, Physical Review Letters 98 (2007), art. no. 230502.
[36] R.Y. Duan, Y. Feng, M.S. Ying, Perfect distinguishability of quantum operations, arXiv:0908.0119 [quant-ph].
[37] E. D'Hondt, P. Panangaden, Quantum weakest preconditions, Mathematical Structures in Computer Science 16 (2006) 429–451.
[38] E. D'Hondt, P. Panangaden, The computational power of the W and GHZ states, Quantum Information and Computation 6 (2006) 173–183.
[39] J. Eisert, M. Wilkens, M. Lewenstein, Quantum games and quantum strategies, Physical Review Letters 83 (1999) 3077.
[40] A. Ezhov, D. Ventura, Quantum neural networks, in: N. Kasabov (Ed.), Future Directions for Intelligent Systems and Information Science, Physica-Verlag, 2000.
[41] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, arXiv:quant-ph/0001106.
[42] E. Farhi, S. Gutmann, Quantum computation and decision trees, Physical Review A 58 (1998) 915–928.
[43] Y. Feng, R.Y. Duan, Z.F. Ji, M.S. Ying, Proof rules for the correctness of quantum programs, Theoretical Computer Science 386 (2007) 151–166.
[44] Y. Feng, R.Y. Duan, Z.F. Ji, M.S. Ying, Probabilistic bisimulations for quantum processes, Information and Computation 205 (2007) 1608–1639.
[45] Y. Feng, R.Y. Duan, M.S. Ying, Unambiguous discrimination between mixed quantum states, Physical Review A 70 (1) (2004), art. no. 012308.
[46] Y. Feng, S.Y. Zhang, R.Y. Duan, M.S. Ying, Lower bound on inconclusive probability of unambiguous discrimination, Physical Review A 66 (2002), art. no. 062313.
[47] R.P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (1982) 467–488.
[48] S. Gay, Quantum programming languages: Survey and bibliography, Mathematical Structures in Computer Science 16 (2006) 581–600.
[49] S.J. Gay, R. Nagarajan, Communicating quantum processes, in: Proceedings of the 32nd ACM Symposium on Principles of Programming Languages, 2005.
[50] A.M. Gleason, Measures on the closed subspaces of a Hilbert space, Journal of Mathematics and Mechanics 6 (1957) 885–893.
[51] D. Gottesman, I. Chuang, Quantum teleportation as a universal computational primitive, Nature 402 (1999) 390–393.
[52] L.K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of 28th Annual ACM Symposium on the Theory of Computing, 1996, p. 212.
[53] L.K. Grover, Quantum telecomputation, arXiv:quant-ph/9704012.
[54] S. Gudder, Quantum automata: An overview, International Journal of Theoretical Physics 38 (1999) 2261–2282.
[55] C. Heunen, B. Jacobs, Quantum logic in dagger kernel categories, in: Proceedings of Quantum Physics and Logic, 2009.

[56] H. Hirsh, A quantum leap for AI, IEEE Intelligent Systems (July/August 1999) 9.
[57] T. Hogg, Quantum search heuristics, IEEE Intelligent Systems (July/August 1999) 12–14.
[58] I.D. Ivanovic, How to differentiate between nonorthonormal states, Physical Letters A 123 (1987) 257–259.
[59] Z.F. Ji, Y. Feng, R.Y. Duan, M.S. Ying, Identification and distance measures of measurement apparatus, Physical Review Letters 96 (2006), art. no. 200401.
[60] P. Jorrand, M. Lalire, Toward a quantum process algebra, in: Proceedings of the 1st ACM Conference on Computing Frontier, ACM Press, 2004.
[61] A. Kitaev, Fault-tolerant quantum computation by anyons, quant-ph/9707021.
[62] E.H. Knill, Conventions for quantum pseudocode, Technical Report LAUR-96-2724, Los Alamos National Laboratory, 1996.
[63] A. Kondacs, J. Watrous, On the power of quantum finite state automata, in: Proceedings of 38th IEEE Conference on Foundations of Computer Science, 1997, pp. 66–75.
[64] K.B. Laskey, Quantum causal networks, in: P.D. Bruza, W. Lawless, C.J. van Rijsbergen, D. Sofge (Eds.), Proceedings of the AAAI Spring Symposium on Quantum Interaction, AAAI Press, Menlo Park, 2007.
[65] D.W. Leung, Quantum computation by measurements, International Journal of Quantum Information 2 (2004) 33–43.
[66] D. Maslov, G.W. Dueck, M. Miller, C. Negrevergne, Quantum circuit simplification and level compaction, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27 (2008) 436–444.
[67] R.V. Meter, K.M. Itoh, Fast quantum modular exponentiation, Physical Review A 71 (2005), art. no. 052320.
[68] K. Miakisza, E.W. Piotrowski, J. Sładkowskic, Quantization of games: Towards quantum artificial intelligence, Theoretical Computer Science 358 (2006) 15–22.
[69] C. Moore, J.P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (2000) 275–306.
[70] S.-C. Mu, R. Bird, Functional quantum programming, in: Proceedings of the 2nd Asian Workshop on Programming Languages and Systems, 2001.
[71] A. Narayanan, M. Moore, Quantum inspired genetic algorithms, in: Proceedings of IEEE International Conference on Evolutionary Computing, 1996, pp. 61–66.
[72] D. Nelson, C.L. McEvoy, L. Pointer, Spreading activation or spooky action at a distance?, Journal of Experimental Psychology: Learning, Memory and Cognition 29 (2003) 42–52.
[73] M.A. Nielsen, Quantum computation by measurement and quantum memory, Physical Letters A 308 (2003) 96–100.
[74] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.
[75] N.J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, 1998.
[76] H. Nishimura, T. Yamakami, An application of quantum finite automata to interactive proof systems, in: M. Domaratzki, A. Okhotin, K. Salomaa, S. Yu (Eds.), The 9th International Conference on Implementation and Application of Automata, Revised Selected Papers, in: LNCS, vol. 3317, Springer, 2005, pp. 225–236.
[77] B. Ömer, Structural quantum programming, Ph.D. thesis, Technical University of Vienna, 2003.
[78] J. Pearl, Causality: Models, Reasoning, and Inference, Cambridge University Press, 2000.
[79] R. Penrose, The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics, Oxford University Press, 1990.
[80] A. Petersen, M. Oskin, A new algebraic foundation for quantum programming languages, in: Proceedings of the 2nd Workshop on Non-Silicon Computing, 2003.
[81] J.F. Poyatos, J.I. Cirac, P. Zoller, Complete characterization of a quantum process: The two-bit quantum gate, Physical Review Letters 78 (1997) 390–393.
[82] J. Preskill, Topological quantum computation, in: Quantum Computation, in: Lecture Notes for Physics, vol. 219, California Institute of Technology, 2004, Chapter 9, http://www.theory.caltech.edu/people/preskill/ph229.
[83] R. Raussendorf, H.J. Briegel, A one-way quantum computer, Physical Review Letters 82 (2001) 5188–5191.
[84] J.W. Sanders, P. Zuliani, Quantum programming, in: Proceedings, Mathematics of Program Construction, in: LNCS, vol. 1837, Springer-Verlag, 2000, pp. 88–99.
[85] M. Sasaki, A. Carlini, Quantum learning and universal quantum matching machine, Physical Review A 66 (2002) 022303.
[86] R. Schack, T.A. Brun, C.M. Caves, Quantum Bayes rule, Physical Review A 64 (2001) 014305.
[87] P. Selinger, Towards a quantum programming language, Mathematical Structures in Computer Science 14 (2004) 527–586.
[88] P. Selinger, B. Valiron, A lambda calculus for quantum computation with classical control, Mathematical Structures in Computer Science 16 (2006) 527–552.
[89] P. Selinger, B. Valiron, On a fully abstract model for a quantum linear functional language, Electronic Notes in Theoretical Computer Science 210 (2008) 123–137.
[90] V.V. Shende, A.S. Bullock, I.L. Markov, Synthesis of quantum-logic circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25 (2006) 1000–1010.
[91] P.W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in: Proceedings of 35th Annual Symposium on Foundations of Computer Science, IEEE Press, Los Alamitos, CA, 1994, pp. 124–134.
[92] P.W. Shor, Why haven't more quantum algorithms been found?, Journal of the ACM 50 (2003) 87–90.
[93] X.M. Sun, S.Y. Zhang, Y. Feng, M.S. Ying, Mathematical nature of and a family of lower bounds for the success probability of unambiguous discrimination, Physical Review A 65 (2002), art. no. 044306.
[94] K.M. Svore, A.V. Aho, A.W. Cross, I.L. Chuang, I.L. Markov, A layered software architecture for quantum computing design tools, IEEE Computer 39 (2006) 74–83.
[95] G. Takeuti, Quantum set theory, in: E. Beltrametti, B.C. van Fraassen (Eds.), Current Issues in Quantum Logics, Plenum, New York, 1981, pp. 303–322.
[96] S. Tani, H. Kobayashi, K. Matsumoto, Exact quantum algorithms for the leader election problem, in: V. Diekert, B. Durand (Eds.), Proc. STACS 2005, in: LNCS, vol. 3404, Springer-Verlag, 2005, pp. 581–592.
[97] A.V. Tonder, A lambda calculus for quantum computation, SIAM Journal on Computing 33 (2004) 1109–1135.
[98] R.R. Tucci, Quantum Bayesian nets, International Journal of Modern Physics B 9 (1995) 295–337.
[99] V.S. Varadarajan, Geometry of Quantum Theory, Springer-Verlag, New York, 1985.
[100] K. Vogel, H. Risken, Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase, Physical Review A 40 (1989) 7113–7120.
[101] G.M. Wang, M.S. Ying, Unambiguous discrimination among quantum operations, Physical Review A 73 (4) (2006), art. no. 042301.
[102] A.C. Yao, Quantum circuit complexity, in: Proc. of the 34th Ann. IEEE Symp. on Foundations of Computer Science, 1993, pp. 352–361.
[103] A. Yimsiriwattana, S.J. Lomonaco, Distributed quantum computing: A distributed Shor algorithm, in: E. Donkor, A.R. Pirich, H.E. Brandt (Eds.), Quantum Information and Computation II, in: Proceedings of SPIE, vol. 5436, 2004, pp. 360–372.
[104] M.S. Ying, A theory of computation based on quantum logic, Theoretical Computer Science 344 (2005) 134–207.
[105] M.S. Ying, Quantum logic and automata theory, in: K. Engesser, D. Gabbay, D. Lehmann (Eds.), Handbook of Quantum Logic and Quantum Structures, Elsevier, Amsterdam, 2007, pp. 619–754.
[106] M.S. Ying, Hoare logic for quantum programs, arXiv:0906.4986 [quant-ph].
[107] M.S. Ying, J.X. Chen, Y. Feng, R.Y. Duan, Commutativity of quantum weakest preconditions, Information Processing Letters 104 (2007) 152–158.

[108] M.S. Ying, R.Y. Duan, Y. Feng, Z.F. Ji, Predicate transformer semantics of quantum programs, in: I. Mackie, S. Gay (Eds.), Semantic Techniques in Quantum Computation, Cambridge University Press, 2009.
[109] M.S. Ying, Y. Feng, An algebraic language for distributed quantum computing, IEEE Transactions on Computers 58 (2009) 728–743.
[110] M.S. Ying, Y. Feng, Algebra of controlled circuits and quantum multiplexors, submitted for publication.
[111] M.S. Ying, Y. Feng, Quantum loop programs, submitted for publication.
[112] M.S. Ying, Y. Feng, R.Y. Duan, Z.F. Ji, An algebra of quantum processes, ACM Transactions on Computational Logic 10 (2009), art. no. 19.
[113] S.Y. Zhang, Y. Feng, X.M. Sun, M.S. Ying, Upper bound for the success probability of unambiguous discrimination among quantum states, Physical Review A 64 (2001), art. no. 062103.