RESEARCH ARTICLE OPEN ACCESS

Computational Quantum Theory & AI

Yew Kee Wong
JiangXi Normal University, Jiangxi - China

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.

Keywords — Quantum Computation, Artificial Intelligence, Autonomy, Cryptography, Privacy Protection

I. 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 classi- cal 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 simulta- neously. 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 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 [55]. 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.

II. QUANTUM COMPUTATION

For convenience of the readers, I will give a very brief introduction to quantum computation in this section. The funda- mental 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 [24].

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:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \quad (1)$$

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:

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

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 [14], 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.

A. Quantum Turing Machine and Quantum Automata

The models of quantum computation have their ancestors from the studies of connections between physics and com- putation. 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 pos- sibility. 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 [23], Gudder [54], and Moore and Crutchfield [39] in the late 1990's. Their definitions of quantum automata differ mainly in where quantum measurements are allowed. For example, a quantum automaton in- troduced in [29] may be observed only after all input symbols have been read, whereas a quantum automaton in [33] is allowed to be observed after reading each symbol. The most general model of quantum finite automata was proposed inde- pendently 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 [36] 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.

B. 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 [22] 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 [40].

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 [37] proposed a compaction method for quantum circuits of modular exponentiation; (2) general techniques; for example, Maslov et al. [26] 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 [19] 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 [11].

C. 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 com- putation 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.

D. Measurement-Based Quantum Computation

Another model of quantum computation without a classical counterpart is measurement-based computation. In the quan- tum Turing machine and quantum circuits, measurements are mainly used at the end to extract computational outcomes from quantum states. However, Raussendorf and Briegel [43] proposed a one-way quantum computer and Nielsen [33] and Leung [45] 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 prepara- tion of the 0 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.

E. Topological Quantum Computation

A crucial challenge in constructing large quantum computers is quantum decoherence. In 1997, topological quantum computation was proposed by Kitaev [31] 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 [22].

F. 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 implemen- tation 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 [13]. Another major motivation comes from the studies of quantum communi- cation. 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 tech- niques for verifying quantum communication protocols, Gay and Nagarajan [49] defined a language CQP (Communicating Quantum Processes) and Jorrand and Lalire [50] 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]. 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 [36] 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.

G. 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 [27], 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 [32] 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.

H. 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. [44] introduced a layered software architecture for quantum computer design tools.

I. Quantum Programming

The earliest proposal for a quantum programming language was made by Knill [22]. The first real quantum programming language, QCL, was proposed by Ömer [37]; 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 [24]. 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 [17] 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 [37]. In [11], 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 [44] in designing qGCL, where quantum computation is reduced to probabilistic computation by the observation (measurement) procedure. Thus, predicate transformer semantics developed for probabilis- tic 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 rep- resentation 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].

III. CONCLUSIONS

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 [37] 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.

REFERENCES

[1] S. Abramsky, High-level methods for quantum computation and information, in: Proceedings of the 19th Annual IEEE Symposium on Logic in Com- puter 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.