量子程序设计研究的近期进展

应明生(University of Technology Sydney, 清华大学)

一. 引言

八年前，我与冯元、段润尧、季铮锋三位老师一起在本刊发表过一篇量子程序理论研究的综述［1］。这些年，这个领域有了比较大的进展，我自己对于一些问题的理解也有所不同。因此，非常高兴得到吴俊杰教授邀请在他主编的专辑中再写一篇这方面的文章。

量子计算机硬件的研制这几年有较大的进展，但目前离实用化还有很大的距离（想必其他老师会在本专辑详加论述）。因此，这篇文章面临的第一个问题是：我们现在研究量子程序是不是太早了？其实这个问题我回答不好，但为了把这篇文章继续写下去，请允许我谈两件事情。首先，早在 1996 年 Knill 已经开始考虑量子程序设计的问题，此后 20 年这方面已经有大量的研究工作发表。第二件事情则扯得有点远。我国正在大力提倡原始创新，而原始创新只有在新领域机会多一点，在成熟的领域则机会很少。量子程序恰好是一个正在兴起的新领域，希望有更多的年轻人参与研究。

当然，量子程序设计研究的是：如何为将来的量子计算机设计程序？文章写到这里则面临第二个问题：既然我们目前并没有实用化的量子计算机，那怎样研究量子程序？惭愧的是这个问题我也回答不好，但本文中我将较为全面地介绍近年来的一些进展。等读者耐心读完本文后，您自己一定能够根据已有的研究工作（以及您认为应该研究而还没有得到研究的问题），比我更好地回答这个问题。由于已有文［1］，本文主要讨论 2008 年以后的工作。

总的来说，到目前为止关于量子程序的研究主要围绕以下两个主题开展［2］：（1）过去为经典计算机发展的程序设计理论、方法和技术如何扩展到量子计算机上？（2）什么样的新程序设计模型、方法和技术能够更有效地发挥量子计算机特有的优势？由于量子系统的一些新特性（体现在量子程序中，如量子数据不可克隆、进程之间的纠缠、量子程序变元可观测量的非交换性），已有的一些程序设计的理论、方法和技术不适用于量子程序，而需要引入一系列全新思想。

二. 量子程序设计语言及其实现

早期的量子程序研究集中在量子程序设计语言的设计及初步的实现。早在 1990 年代及 2000 年代初就已经提出了几个高级量子程序设计语言，这方面的简单介绍可参见［1］。

最近这几年，由于受到量子计算硬件进展的刺激以及 2010 年 IARPA 设立的量子计算机科学项目的推动，量子程序设计语言的设计与实现方面有很大的进展，主要包括：（1）Microsoft 的 LIQUi|>［3］。最近他们与 ETH Zurich 合作进一步地提出了量子程序编译和优化的可扩展软件设计流［4］；(2) Selinger 组的 Quipper ［5］；(3) Princeton、UCSB 等单位的 Scaffold［6, 7］。(4) Raytheon BBN Technologies 等的 QuaFL[8, 9]。特别值得指出的是，为了更好地实现其编译器，Microsoft 与 Selinger 两个研究组在量子电路的优化方面做出了一些非常好的工作，较好地解决了单个量子比特电路优化问题，但多个量子比特电路的优化进展不大。量子电路的优化远较经典电路优化困难得多，经典电路的优化方法完全不适用，而需要一些很不一样的思路，比如单量子比特情形采用了一些初等代数数论工具。随便提及，布尔代数为经典电路的分析与综合提供一种很好的代数语言。文［10］曾为量子电路定义了一种代数语言，以便对量子电路进行代数操作与推理。

在国内，南京大学徐家福、宋方敏、吴楠教授研究组在量子程序设计语言及其实现方面做了大量的工作，在［11］第 8 节有一个很好的介绍。文［12］也曾定义了一个量子 flowchart 语言。

三．量子程序的语义

程序设计语言的形式语义为程序提供了严格的数学描述。这样的描述对于量子程序尤为重要，因为量子系统的一些特征违反人们的直觉，理解起来很容易出错，而对于量子力学可能缺乏系统训练的程序员则尤其如此。

量子程序语义的早期研究在［1］中已有介绍。这里只谈近几年的一些工作。基于 Girard 的 Geometry of Interaction (Abramsky-Haghverdi-Scott 范畴论形式)，Hasuo 与 Hoshino［13］找到了带递归的函数式量子程序设计语言的一个语义模型。进一步地，利用线性逻辑定量语义中的一些构造，Pagani, Selinger and Valiron［14］定义了带递归及无限数据类型的函数式量子程序设计语言的一个指称语义。Jacobs［15］为量子程序的 block 构造给出了一个范畴论公理化。Staton［16］为量子程序的等式推理提供了一个代数语义框架。

在量子程序的最弱前置条件语义的研究中，早在 2006 年 D’Hondt 与 Panangaden［17］就提出了将量子谓词定义为本征值在单位区间中的物理可观测量（Hermitian 算子）。这是一个一旦说出来以后大家都觉得极其简单的想法，但对于进一步的发展却很重要。文［18］考虑了一类特殊的量子谓词－投影算子，从而使得量子逻辑中发展起来的一些方法可应用于量子程序的谓词转换器语义。需要特别指出的是，量子程序最弱前置条件语义的研究中存在着一个特有（经典程序所没有）的困难：对于非交换的量子谓词（Hermitian 算子），其析取、合取无法定义。

四．量子程序的分析

程序分析技术在编译器设计及程序优化等方面有着重要的应用。在 Scaffold［6］的编译器 ScaffCC［7］中已经采用了数据流分析的一些方法，分析量子程序中的纠缠并检查是否违反不可克隆原理。文［7］也考虑了执行时间估计的问题。

文［19］首先研究了有限维 Hilbert 空间中循环体(loop body)为 unitary 算子的量子循环程序的终止问题。文［20］将［19］中的一些主要结果推广到循环体为一般超算子(super-operator)的情形，并考虑了平均执行时间的问题。这项工作中采用的主要技术是超算子的矩阵表示及 Schroedinger-Heisenberg 对称性。按照［20］提出的将量子 Markov 链作为量子程序语义模型的想法，量子程序的终止问题可以看作量子 Markov 链的一类可达性问题。因此，应圣刚、李杨佳等在［21，22］中仔细研究了量子 Markov 链及量子自动机的可达性。值得一提的是，Markov 链的可达性分析往往归结为图论中的相应问题。但这些图论工具并不适用于量子 Markov 链，因此作为量子 Markov 链可达性分析所需的数学工具，文［21］初步建立了一种 Hilbert 空间中的新图论。这种新的图论本身似乎就很值得研究。

Perdrix 与 Jorrand［23， 24］将抽象解释（abstract interpretation）技术引入量子程序分析，特别是程序中的纠缠及其演化的分析。Honda［25］进一步地分析了可用 stabilizer formalism 描述的这一类特殊量子程序中的纠缠。量子程序中的纠缠分析看来是一个重要的问题，但文［7］与［23, 24, 25］中的工作还是非常初步的，希望有更加深刻的研究。

五．量子程序的验证

如第三节所述，量子系统的许多特征违反人们的直觉。因此，将来的量子程序设计员可能更容易犯错误。这就使得量子程序的验证成为一个重要的问题。

量子程序验证研究的一个主要的方面是发展适用于量子计算的程序逻辑。Brunet 与 Jorrand［26］提出了将 Birkhoff-von Neumann 量子逻辑应用于量子程序推理的方法。Baltag 与 Smets［27］发展了一种可用于量子系统中信息流形式化的动态逻辑。在只允许有界迭代的限制下，Chadha, Mateus 与 Sernadas［28］给出了量子程序的一个 Floyd-Hoare 型证明系统。冯元等［29］发现了关于量子程序的一些有用的推理规则。Kakutani［30］试图将 den Hartog 的概率 Hoare 逻辑推广到量子情形。文［31］建立了一个真正完整的量子 Floyd-Hoare 逻辑，特别是证明了其（相对）完备性。至此，顺序量子程序的逻辑基础已经建立起来了。但分布式、并发量子计算的程序逻辑还完全处于空白状态(其重要性参见下一节)。

在量子程序验证工具方面，Anticoli［32］将 Quipper 翻译到超算子值（量子）Markov 链［33, 34］，然后用针对这类 Markov 链的模型检测工具 QPMC［35］验证。最近，中国科学院软件研究所詹乃军教授研究组基于 Isabelle/HOL 设计实现了一个量子 Floyd-Hoare 逻辑定理证明器。为了提高其自动化程度，有必要研究量子程序不变量与 ranking 函数的生成算法。有一段时间，我们甚至不懂得如何定义量子程序的不变量。最近，我们找到了解决这个问题方法，并将量子程序不变量生成问题转换为半正定规划(Semidefinite Programming)问题。对于 ranking 函数的生成，现有的数学工具似乎不足以解决这个问题，我们可能需要自行发展必要的数学工具，特别是量子 martigale 及 supermartigale 理论。

五．量子通讯并发系统

量子通讯并发系统研究的主要动因有二。首先，普遍认为量子通讯的实用化会比量子计算机早很多，而量子通讯并发系统的研究可能为量子通讯协议的设计、分析与验证提供理论基础和工具。其次，大型的量子计算机还非常遥远，但最近美国总统科学技术办公室发布的量子信息文件预计几十个量子比特、可供早期量子计算机科学研究的系统可望在5年内出现［36］。因此，一个自然的想法是如何将多个小型量子计算机组成分布式系统解决经典计算机所不能解决的问题。实际上，十多年前物理学家就已经开始研究分布式量子计算。今年初Bill Gates [37]也曾预测量子计算云服务将在十年内出

现。这样，并发问题在分布式量子程序设计中不可避免。

目前关于量子通讯并发系统的工作主要是量子进程代数及其在量子通讯协议验证中的应用。Jorrand与Lalire [38]、 Gay与Nagarajan [39]、冯元等［40, 41］分别定义了量子进程代数 QPAlg、CQP、qCCS。一个困难的问题在于找到有纠缠的并行复合保持的互模拟，这个问题是量子情形特有的，因为在经典进程代数中没有纠缠。冯元等在［42，43］中解决了这个问题。考虑到量子计算与通讯中量子噪音是一个不可忽略的因素，文［41, 43］还研究了量子进程的近似互模拟。基于qCCS, Kubota［44, 45］设计实现了量子密码协议验证的软件工具。

此外，Gay等［46］、冯元等［33, 35］以及文［47］将模型检测(model-checking)技术分别推广到可用stabilizer formalism描述的特殊量子系统以及一般的量子系统。Ardeshir-Larijani等［48, 49］研究了量子通讯协议的等价性检测。 俞能昆等［50］考虑了公平条件（fairness）下并发量子程序的终止问题。目前这方面的研究尚只涉及量子系统的一些比较简单的性质（如可达性）。如果希望分析或验证量子程序的更复杂的性质，一个首先必须解决的问题是定义一种可以表达这些性质的时序逻辑。由于量子测量会改变量子系统的状态，这样的时序逻辑的定义并不是一个容易的问题。事实上，一些物理学家在20多年前为了别的目的已经做过一些尝试，但并没有完全成功。另一个重要的问题是如何验证量子性质（如纠缠）。最近许多物理学文献讨论这个问题，但如何采用计算机科学的工具更好地解决这个问题尚未得到考虑。

需要特别指出的是，虽然关于量子通信并发系统已经有不少的研究工作发表，但我们对于量子纠缠、nonlocality、contextuality 等量子特征在这些系统中的作用与影响还没有很好的理解。

六． 从“数据叠加”到“程序叠加”

前面五节中论及的量子程序研究都遵循“程序的数据流是量子的，但控制流仍然是经典的”这样一个基本的思路。Selinger 将这个思路总结为一句口号：“量子数据，经典控制（quantum data, classical control）”。这个思路的提出是十分自然的，也是相对来说比较容易实现的，只需要在经典的程序设计语言中增加对于量子数据的操作，如 unitary 变换, 量子测量等。

但是，人们也认识到“量子数据，经典控制”的思想并不能最为有效地发挥量子计算特有的优势。Altenkirch 与 Grattage［51］

提出了一个带量子控制流的函数式量子程序设计语言 QML，并在其后一系列论文中系统地发展了相应的理论，甚至实现了编译器。但是，现在看来这个理论并没有完全成功。直到最近，受到量子随机游走（quantum random walks）的启发，书［2］第 6、7 章找到了一种真正能实现量子控制流的量子程序设计模型。在没有递归（也没有循环）的情况下，如果整个程序中测量都可以移到最后一步，带量子控制流的程序的语义容易定义，本质上就是量子的 multiplexor。如果中间过程有量子测量，这个问题比较困难。书［2］第 6 章通过引入算子值函数的卫式(Guarded)复合，进一步定义超算子的卫式复合，从而由量子程序的半经典语义导出纯量子语义解决这一问题。Badescu 与 Panangaden 在［52］中对这个问题有一些有趣的讨论。正如［52］中指出的，带量子控制流的量子递归是一个非常困难的问题，需要全新的思想。书［2］第 6 章提出采用二次量子化方法（second quantization）解决这个问题, 因而定义了波色子（对称的）与费米子（反对称）量子递归模式。必须承认的是，我们对于量子递归还远没有完全彻底地理解。

如本节一开始所指出的，第五节讨论的量子通讯并发系统不带量子控制流。如果将量子控制流引入通讯并发系统，并结合物理学中发现的一些量子同步机制（synchronization）[53], 问题将变得极为复杂，似乎超出目前我们能够理解的范围。

七． 结束语：上帝是伟大的量子程序设计员？

如前所述，带着实用化量子计算机即将到来的期望，研究人员已经在量子程序方面做了大量的工作。但是在结束本文之前，一个不可避免的问题是：如果实用化量子计算机永远无法实现，那这些工作不就是无用功吗？我没能很好地回答引言中的两个问题。遗憾的是，我也不太会回答这个问题，只能试一试。

乐观的回答：许多物理学家认为量子计算机的建造在原理上已经没有障碍，主要的困难在于工程技术层面。在第五节中已经谈及，美国总统科学技术办公室最近发布的量子信息文件预计几十个量子比特的量子计算机可望在 5 年内出现［36］。因此，我们或可乐观地期望量子程序设计的研究成果能够在 10－20 年内得到实际的应用。

保守的回答：人们普遍认为，我们正处于第二次量子革命［54］：从量子物理到量子工程技术，特别是量子信息技术。量子物理的目的在于发现存在于自然中的规律，而量子工程技术的目的

则是基于量子物理原理设计实现新的器件和系统，完成基于经典物理的技术无法完成的任务。因此，即使量子计算机难以实用化，量子程序设计理论、方法和技术也能够在这些领域得到直接的应用（如嵌入式量子芯片的测试与验证）或起到借鉴的作用。

更富想象力的回答：近年来，许多计算机科学家提倡computational thinking [55]。其实最大胆的 computational thinking 是物理学家做出的。一些大物理学家[56, 57]提出了如下假设：宇宙是一台（巨型）量子计算机。如果你接受这个假设的话，也许你会乐意接受我进一步的假设：（斯宾诺莎的）上帝是一位量子程序设计员！在这样的假设下，将程序设计理论中的一些思想引入量子物理学，有可能为物理学提供一个新的视角。

致谢：衷心感谢我们课题组的所有成员，16 年的教学相长使我受益匪浅。特别感谢课题组早期成员冯元、段润尧、季铮锋，16 年后我们仍然在一起愉快地合作以及午餐时聊着学术八卦。

参考文献

［1］冯元，段润尧，季铮锋，应明生，量子程序理论及相关问题研究，中国计算机学会通讯，2008, 7.

［2］M. S. Ying, Foundations of Quantum Programming, Elsevier – Morgan Kaufmann, 2016.

［3］D. Wecker and K. M. Svore, LIQUi|>: A software design architecture and domain-specific language for quantum computing, http://research.microsoft.com/pubs/209634/1402.4467.pdf.

［4］T. Haener, D. S. Steiger, K. Svore and M. Troyer, A software methodology for compiling quantum programs, *arXiv:*1604.01401v1

［5］A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger and B. Valiron, Quipper: A scalable quantum programming language, Proceedings of the 34th ACM Conference on Programming Language Design and Implementation (PLDI), 2013, pp. 333-342.

［6］A. J. Abhari, A. Faruque, M. Dousti, L. Svec, O. Catu, A. Chakrabati, C.-F. Chiang, S. Vanderwilt, J. Black, F. Chong, M. Martonosi, M. Suchara, K. Brown, M. Pedram and T.Brun, Scaffold: Quantum Programming Language, Technical Report TR-934-12, Dept. of Computer Science, Princeton University, 2012.

［7］A. JavadiAbhari, S. Patil, D. Kudrow, J. Heckey, A. Lvov, F. T. Chong and M. Martonosi, ScaffCC: Scalable compilation and analysis of quantum programs, Parallel Computing, 45(2015)2-17.
［8］A. Lapets and M. Roetteler, Abstract resource cost derivation for logical quantum circuit description, Proceedings of the ACM Workshop on Functional Programming Concepts in Domain-Specific Languages (FPCDSL), 2013, pp. 35-42.
［9］A. Lapets, M. P. da Silva, M. Thome, A. Adler, J. Beal and M. Roetteler, QuaFL: A typed DSL for quantum programming, Proceedings of the ACM Workshop on Functional Programming Concepts in Domain-Specific Languages (FPCDSL), 2013, pp. 19-27.
［10］M. S. Ying and Y. Feng, An algebraic language for distributed quantum computing, IEEE Transactions on Computers，58(2009)728-743.
［11］吴楠，宋方敏，X. D. Li, 通用量子计算机：理论、组成与实现，计算机学报，38：14（2015）1-19.
［12］M. S. Ying and Y. Feng, A flowchart language for quantum programming, IEEE Transactions on Software Engineering, 37(2011)466-485.
［13］I. Hasuo and N. Hoshino, Semantics of higher-order quantum computation via Geometry of Interaction, Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS), 2011, pp. 237-246.
［14］M. Pagani, P. Selinger and B. Valiron, Applying quantitative semantics to higher-order quantum computing, Proceedings of the 41st ACM Symposium on Principles of Programming Languages (POPL), 2014, pp. 647-658.
［15］B. Jacobs, On block structures in quantum computation, Electronic Notes in Theoretical Computer Science，298(2013)233-255.
［16］S. Staton, Algebraic effects, linearity, and quantum programming languages, Proceedings of the 42nd ACM Symposium on Principles of Programming Languages (POPL), 2015, pp. 395-406.
［17］E. D'Hondt and P. Panangaden, Quantum weakest preconditions, Mathematical Structures in Computer Science, 16(2006)429-451.
［18］M. S. Ying, R. Y. Duan, Y. Feng and Z. F. Ji, Predicate transformer semantics of quantum programs, In: Semantic Techniques in Quantum Computation （I. Mackie and S. Gay, eds.）, Cambridge University Press 2010, pp. 311-360.

［19］M. S. Ying and Y. Feng, Quantum loop programs, Acta Informatica, 47(2010), 221-250.
［20］M. S. Ying, N. K. Yu, Y. Feng and R. Y. Duan, Verification of quantum programs, Science of Computer Programming, 78(2013)1679-1700.
［21］S. G. Ying, Y. Feng, N. K. Yu and M. S. Ying, Reachability analysis of quantum Markov chains, Proceedings of the 24th International Conference on Concurrency Theory (CONCUR), 2013, pp. 334-348.
［22］Y. J. Li and M. S. Ying, (Un)decidable problems about reachability of quantum systems, Proceedings of the 25th International Conference on Concurrency Theory (CONCUR), 2014, pp. 482-496.
［23］S. Perdrix, Quantum entanglement analysis based on Abstract Interpretation, Proceedings of the 15th International Symposium on Static Analysis (SAS), 2008, pp. 270-282.
［24］P. Jorrand and S. Perdrix, Abstract interpretation techniques for quantum computation, In: Semantic Techniques in Quantum Computation（I. Mackie and S. Gay, eds.）, Cambridge University Press 2010, pp. 206-234.
［25］K. Honda, Analysis of quantum entanglement in quantum programs using stabiliser formalism, Electronic Proceedings in Theoretical Computer Science 195 (QPL 2015), pp. 262-272.
［26］O. Brunet and P. Jorrand, Dynamic quantum logic for quantum programs, International Journal of Quantum Information, 2(2004)45-54.
［27］A. Baltag and S. Smets, LQP: the dynamic logic of quantum information, Mathematical Structures in Computer Science, 16(2006)491-525.
［28］R. Chadha, P. Mateus and A. Sernadas, Reasoning about imperative quantum programs, Electronic Notes in Theoretical Computer Science, 158(2006)19-39.
［29］Y. Feng, R. Y. Duan, Z. F. Ji and M. S. Ying, Proof rules for the correctness of quantum programs, Theoretical Computer Science, 386(2007)151-166.
［30］Y. Kakutani, A logic for formal verification of quantum programs, Proceedings of the 13th Asian Computing Science Conference (ASIAN 2009), pp. 79-93.

［31］M. S. Ying, Floyd-Hoare logic for quantum programs, ACM Transactions on Programming Languages and Systems， 39(2011), art. no. 19.
［32］L. Anticoli, C. Piazza, L. Taglialegne and P. Zuliani, Towards quantum programs verification: from Quipper circuits to QPMC, Proceedings of the 8th International Conference on Reversible Computation (RC), 2016, pp. 213-219.
［33］Y. Feng, N. K. Yu and M. S. Ying, Model checking quantum Markov chains, Journal of Computer and System Sciences, 79(2013)1181-1198.
［34］Y. Feng, N. K. Yu and M. S. Ying, Reachability analysis of recursive quantum Markov chains, Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS), 2013, pp. 385-396.
［35］Y. Feng, E. M. Hahn, A. Turrini and L. J. Zhang, QPMC: a model checker for quantum programs and protocols, Proceedings of the 20th International Symposium on Formal Methods (FM 2015), Springer LNCS 9109, pp. 265-272.
［36］Advancing Quantum Information Science: National Challenges and Opportunities, https://www.whitehouse.gov/sites/whitehouse.gov/files/images/Quantum_Info_Sci_Report_2016_07_22%20final.pdf
［37］Bill Gates: The future of quantum computing is bright, http://pointbandbeyond.com/news/bill-gates-the-future-of-quantum-computing-is-bright/
［38］P. Jorrand and M. Lalire, Toward a quantum process algebra, Proceedings of the 1st ACM Conference on Computing Frontier, 2004, pp. 111-119.
［39］S. J. Gay and R. Nagarajan, Communicating Quantum Processes, Proceedings of the 32nd ACM Symposium on Principles of Programming Languages (POPL), 2005, pp. 145-157.
［40］Y. Feng, R. Y. Duan, Z. F. Ji and M. S. Ying, Probabilistic bisimulations for quantum processes, Information and Computation, 205(2007)1608-1639.
［41］M. S. Ying, Y. Feng, R. Y. Duan and Z. F. Ji, An algebra of quantum processes, ACM Transactions on Computational Logic, 10(2009), art. no. 19.
［42］Y. Feng, R. Y. Duan and M. S. Ying, Bisimulation for quantum processes, Proceedings of the 38th ACM Symposium on Principles of Programming Languages (POPL), 2011, pp. 523-534.

［43］ Y. Feng, R. Y. Duan and M. S. Ying, Bisimulation for quantum processes, ACM Transactions on Programming Languages and Systems, 34(2012) art. no: 17.
［44］ T. Kubota, Verification of Quantum Cryptographic Protocols using Quantum Process Algebras, PhD Thesis, Department of Computer Science, University of Tokyo, 2014.
［45］ T. Kubota, Y. Kakutani, G. Kato, Y. Kawano and H. Sakurada, Semi-automated verification of security proofs of quantum cryptographic protocols, Journal of Symbolic Computation, 73(2016)192-220.
［46］ S. J. Gay, N. Papanikolaou and R. Nagarajan, QMC: a model checker for quantum systems， Proceedings of the 20th International Conference on Computer Aided Verification (CAV), 2008, pp. 543-547.
［47］ M. S. Ying, Y. J. Li, N. K. Yu and Y. Feng, Model-checking linear-time properties of quantum systems, ACM Transactions on Computational Logic, 15(2014), art. no. 22.
［48］ E. Ardeshir-Larijani, S. J. Gay and R. Nagarajan, Equivalence checking of quantum protocols, Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2013, pp. 478-492.
［49］ E. Ardeshir-Larijani, S. J. Gay and R. Nagarajan, Verification of concurrent quantum protocols by equivalence checking, Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2014), pp. 500-514.
［50］ N. K. Yu and M. S. Ying, Reachability and termination analysis of concurrent quantum programs, Proceedings of the 23th International Conference on Concurrency Theory (CONCUR), 2012, pp. 69-83.
［51］ T. Altenkirch and J. Grattage, A functional quantum programming language, Proc. of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS), 2005, pp. 249-258.
［52］ C. Badescu and P. Panangaden, Quantum alternation: prospects and problems, Electronic Proceedings in Theoretical Computer Science 195 (QPL 2015), pp. 33-42.
［53］ V. Giovannetti, S. Lloyd and L. Maccone, Quantum-enhanced positioning and clock synchronisation, Nature, 412(2001)417-419.
［54］ J. P. Dowling and G. J. Milburn, Quantum technology: the second quantum revolution, Philosophical Transactions of the Royal Society London A, 361(2003)1655-1674.
［55］ J. M. Wing, Computational thinking, Communications of the ACM, 49:3(2006)33-35.

[56] G. ’t Hooft, The cellular automaton interpretation of quantum mechanics – A view on the quantum nature of our universe, compulsory or impossible?, *arXiv*:1405.1548v2.

［57］ S. Lloyd, A theory of quantum gravity based on quantum computation, *arXiv*:quant-ph/0501135