Automated Quantum Program Verification in Probabilistic Dynamic Quantum Logic

Canh Minh Do[1] Tsubasa Takagi[2] Kazuhiro Ogata[1]

[1]Japan Advanced Institute of Science and Technology, 1-8 Asahidai, Nomi, Japan

[2]Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, Japan

Presented Physically at the 2$^{\text{nd}}$ International Workshop on Formal Analysis and Verification of Post-Quantum Cryptographic Protocols

November 21, 2023 - Brisbane, Australia

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Introduction

- Quantum computing is a rapidly emerging technology that uses the laws of quantum mechanics to solve complex problems beyond the capabilities of classical computers, such as Shore's fast algorithms[1] for discrete logarithms and factoring.
- Due to radically different principles of quantum mechanics, such as superposition, entanglement, and measurement, it is challenging to accurately design and implement quantum algorithms, quantum programs, and quantum protocols.
- Therefore, it is crucial to ensure the correctness of quantum systems through verification.

---

[1] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994.

Formal Verification of Quantum Programs

- Previous Studies of Quantum Program Verification
  - Quantum Hoare Logic (QHL)[2]: a quantum counterpart of Hoare Logic
  - Dynamic Quantum Logic (DQL)[3]: a quantum counterpart of Dynamic Logic
- Problems of Previous Studies
  - QHL can semi-automatically perform proofs of correctness with a support tool[4] implemented in Coq. Meanwhile, DQL still requires manual proof verification.
  - In this study, we propose an automatic verification method based on Probabilistic Dynamic Quantum Logic (PDQL), an extended version of Basic Dynamic Quantum Logic (BDQL)[5].

[2] Mingsheng Ying. "Floyd–Hoare Logic for Quantum Programs". In: *ACM Trans. Program. Lang. Syst.* (2012).

[3] Alexandru Baltag and Sonja Smets. "Reasoning about Quantum Information: An Overview of Quantum Dynamic Logic". In: *Applied Sciences* (2022).

[4] Junyi Liu et al. "Formal Verification of Quantum Algorithms Using Quantum Hoare Logic". In: *Computer Aided Verification*. 2019.

[5] Tsubasa Takagi, Canh Minh Do, and Kazuhiro Ogata. "Automated Quantum Program Verification in a Dynamic Quantum Logic". In: *DaLí: Dynamic Logic – New trends and applications*. 2023.

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Hilbert Spaces

- A Hilbert space $\mathcal{H}$ usually serves as the state space of a quantum system that is a complex vector space equipped with an inner product such that each Cauchy sequence of vectors has a limit.
- An n-qubit system is the complex 2^n-space $\mathbb{C}^{2^n}$, where $\mathbb{C}$ stands for the complex plane.
- Pure states in the n-qubit systems $\mathbb{C}^{2^n}$ are unit vectors in 2^n-space $\mathbb{C}^{2^n}$.
- The orthogonal basis called computational basis in the one-qubit system $\mathbb{C}^2$ is the set $\{|0\rangle, |1\rangle\}$ that consists of the column vectors $|0\rangle = (1, 0)^T$ and $|1\rangle = (0, 1)^T$, where ^{T} denotes the transpose operator.
- In the two-qubit system $\mathbb{C}^4$, there are pure states that cannot be represented in the form $|\psi_1\rangle \otimes |\psi_2\rangle$ and called entangled states, where $\otimes$ denotes the tensor product (more precisely, the Kronecker product).
- For example, the EPR state (Einstein-Podolsky-Rosen state) $|EPR\rangle = (|00\rangle + |11\rangle)/\sqrt(2)$ is an entangled state, where $|00\rangle = |0\rangle \otimes |0\rangle$ and $|11\rangle = |1\rangle \otimes |1\rangle$.

Unitary Operators

- Quantum computation is represented by unitary operators (also called quantum gates).
- For example, the Hadamard gate H and Pauli gates X, Y, and Z are quantum gates on the one-qubit system $\mathbb{C}^2$ and are defined as follows:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

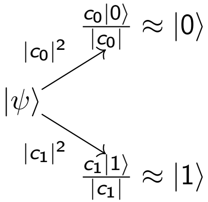
- Two typical quantum gates on the two-qubit systems $\mathbb{C}^4$ are the controlled-X gate (also called the controlled-NOT gate) CX and the swap gate $SWAP$ are defined by

$$CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X,$$
$$SWAP = CX(I \otimes |0\rangle\langle 0| + X \otimes |1\rangle\langle 1|)CX,$$

where I denotes the identity matrix of size 2×2.

Measurement

- Measurement is a completely different process from applying quantum gates. Here we roughly explain specific projective measurements.
- For the general definition of projective measurement, see the famous textbook of quantum computation[6].
- Observe that $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ are projectors, respectively.
- After executing the measurement $\{P_0, P_1\}$, a current state $|\psi\rangle = c_0\,|0\rangle + c_1\,|1\rangle$ is collapsed into either $\frac{P_0|\psi\rangle}{|c_0|}$ with probability $|c_0|^2$ or into $\frac{P_1|\psi\rangle}{|c_1|}$ with probability $|c_1|^2$.



[6] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Regular Program

Program	Name	Meaning
skip	Skip	Do nothing.
abort	Abort	Forcing to halt.
$a\ ;\ b$	Composition	Execute a and then execute b.
$a \cup b$	Non-deterministic Choices	Execute either a or b non-deterministically.
a^*	Iteration	Repeat a some finite number of times.
$p?$	Test	Confirm that p is whether true or false.

- Regular Program = Regular Expression + Test
- Conditional/Loop program consists of regular programs
  - if A then a else b fi $= (A?\ ;\ a) \cup (\neg A?\ ;\ b)$
  - if $A_1 \rightarrow a_1 | \ldots | A_n \rightarrow a_n$ fi $= (A_1?\ ;\ a_1) \cup \ldots \cup (A_n?\ ;\ a_n)$
  - while A do a od $= (A?\ ;\ a)^*\ ;\ \neg A?$
  - repeat a until $A = a\ ;\ (\neg A?\ ;\ a)^*\ ;\ A?$

Dynamic Logic

- Dynamic Logic = Formulas + Regular Programs + Dynamic Operator $[a]$
- The set L of all formulas and the set Π of all regular programs are defined by the following simultaneous induction:

$$L \ni A ::= p \mid \neg A \mid A \wedge A \mid [a]A,$$
$$\Pi \ni a ::= \text{skip} \mid \text{abort} \mid \pi \mid a\,;\,a \mid a \cup a \mid A?,$$

where p denotes an atomic formula and π denotes an atomic program.

Formula	**Name**	**Meaning**
$\neg A$	Negation	Not A
$A \wedge B$	Conjunction	A and B
$[a]A$	Dynamic Operator	It is always A after a is executed

☞ Dynamic Logic is compatible with formal verification because it can express exhaustive searches.

Semantics of DQL

- For the sake of simplicity, we use regular programs Π^- without the iteration operator *.

Definition 1

Quantum dynamic frame is a pair $(\mathcal{H}, v)$ of a Hilbert space $\mathcal{H}$ and a function v from the set Π_0 of all atomic programs to the set $\mathcal{U}(\mathcal{H})$ of all unitary operators on $\mathcal{H}$. Here, v is called an interpretation function of atomic programs.

Definition 2

Quantum dynamic model is a triple $(\mathcal{H}, v, V)$ that consists of a quantum dynamic frame $(\mathcal{H}, v)$ and a function V from the set L_0 of all atomic formulas to the set $\mathcal{C}(\mathcal{H})$ of all closed subspaces of $\mathcal{H}$. Here, V is called an interpretation function of atomic formulas.

- Quantum logic interprets formulas as closed subspaces.

Semantics of DQL

For each quantum dynamic model $M = (\mathcal{H}, v, V)$, the function $[\![\ ]\!]^M : L \to \mathcal{C}(\mathcal{H})$ and family $\{R_a^M : a \in \Pi^-\}$ of relations on $\mathcal{H}$ are defined by simultaneous induction as follows:

1. $[\![p]\!]^M = V(p)$;
2. $[\![\neg A]\!]^M$ is the orthogonal complement of $[\![A]\!]^M$;
3. $[\![A \wedge B]\!]^M = [\![A]\!]^M \cap [\![B]\!]^M$;
4. $[\![[a]A]\!]^M = \{s \in \mathcal{H} : (s,t) \in R_a^M$ implies $t \in [\![A]\!]^M$ for any $t \in \mathcal{H}\}$;
5. $R_{\text{skip}}^M = \{(s,t) : s = t\}$;
6. $R_{\text{abort}}^M = \emptyset$;
7. $R_\pi^M = \{(s,t) : (v(\pi))(s) = t\}$;
8. $R_{a;b}^M = \{(s,t) : (s,u) \in R_a^M$ and $(u,t) \in R_b^M$ for some $u \in \mathcal{H}\}$;
9. $R_{a \cup b}^M = R_a^M \cup R_b^M$;
10. $R_{A?}^M = \{(s,t) : P_{[\![A]\!]^M}(s) = t\}$, where $P_{[\![A]\!]^M}$ stands for the projection onto $[\![A]\!]^M$.

Semantics of DQL

- Henceforth, we write $(M, s) \models A$ for $s \in [\![A]\!]^M$.
- $(M, s) \models A$ if and only if $P_{[[A]]^M}(s) = s$.
  ☞ There is a bijection between a closed subspace and a projection onto it.

Theorem 1

For any M and $s \in \mathcal{H}$, the following holds:

1. $(M, s) \models A \wedge B$, *if and only if* $(M, s) \models A$ *and* $(M, s) \models B$.
2. $(M, s) \models [\mathrm{skip}]A$ *if and only if* $(M, s) \models A$.
3. $(M, s) \models [\mathrm{abort}]A$.
4. $(M, s) \models [\pi]A$ *if and only if* $(M, (v(\pi))(s)) \models A$.
5. $(M, s) \models [a\ ;\ b]A$ *if and only if* $(M, s) \models [a][b]A$.
6. $(M, s) \models [a \cup b]A$ *if and only if* $(M, s) \models [a]A \wedge [b]A$.
7. $(M, s) \models [A?]B$ *if and only if* $(M, P_{[\![A]\!]^M}(s)) \models B$.

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Probabilistic Dynamic Quantum Logic (PDQL)

- To capture the probabilistic ingredient from measurement, we introduce a probabilistic operator $\mathsf{P}^{\geq r}$ to formulate Probabilistic Dynamic Quantum Logic (PDQL) as follows:

$$L \ni A ::= p \mid \neg A \mid A \wedge A \mid [a]A \mid \mathsf{P}^{\geq r}A,$$
$$\Pi \ni a ::= \mathrm{skip} \mid \mathrm{abort} \mid \pi \mid a \,;\, a \mid a \cup a \mid A?,$$

where r denotes a rational number in the closed interval $[0, 1]$.

Formula	**Meaning**
$\mathsf{P}^{\geq r}A$	a projective measurement of A on the current state of a quantum system will succeed with probability $\geq r$.
$[A?^{\geq r}]B \triangleq \mathsf{P}^{\geq r}A \wedge [A?]B$	if the quantum test $A?$ succeeds with probability $\geq r$, then B will be the case after the successful execution of the quantum test.

- Similarly, we can define other probabilistic operators $\mathsf{P}^{>r}$, $\mathsf{P}^{\leq r}$, $\mathsf{P}^{<r}$, $\mathsf{P}^{=r}$, and $\mathsf{P}^{\neq r}$.

Semantics of PDQL

- The function $[\![\ ]\!]^M : L \to \mathcal{C}(\mathcal{H})$ is extended to handle the probabilistic operator $\mathrm{P}^{\geq r}$ using the Born rule as follows:

$$s \in [\![\mathrm{P}^{\geq r} A]\!]^M \text{ if and only if } \left\langle s \middle| P_{[\![A]\!]^M}(s) \right\rangle \geq r,$$

- Henceforth, we write $(M, s) \models \mathrm{P}^{\geq r} A$ if and only if $s \in [\![\mathrm{P}^{\geq r} A]\!]^M$.

Theorem 2

For any M, $s \in \mathcal{H}$, and $r \in [0, 1]$, the following holds:

1. $(M, s) \models [A?^{\geq r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{\geq r} A$ *and* $(M, s) \models [A?]B$.
2. $(M, s) \models [A?^{> r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{> r} A$ *and* $(M, s) \models [A?]B$.
3. $(M, s) \models [A?^{\leq r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{\leq r} A$ *and* $(M, s) \models [A?]B$.
4. $(M, s) \models [A?^{< r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{< r} A$ *and* $(M, s) \models [A?]B$.
5. $(M, s) \models [A?^{= r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{= r} A$ *and* $(M, s) \models [A?]B$.
6. $(M, s) \models [A?^{\neq r}]B$, *if and only if* $(M, s) \models \mathrm{P}^{\neq r} A$ *and* $(M, s) \models [A?]B$.

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Standard Interpretation

- Now we discuss the verification of concrete quantum programs based on PDQL
- Fix Π_0 and L_0 as follows ($\mathbb{N}$ denotes natural numbers including 0 and $\mathbb{C}$ denotes complex numbers):

$$\Pi_0 = \{\mathtt{H}(i), \mathtt{X}(i), \mathtt{Y}(i), \mathtt{Z}(i), \mathtt{CX}(i,j), \mathtt{SWAP}(i,j) : i,j \in \mathbb{N}, i \neq j\},$$
$$L_0 = \{p(i, |\psi\rangle), p(i, i+1, |\Psi\rangle) : i \in \mathbb{N}, |\psi\rangle \in \mathbb{C}^2, |\Psi\rangle \in \mathbb{C}^4\},$$

- Standard interpretation $\bar{v} : \Pi_0 \to \mathcal{U}(\mathbb{C}^{2^n})$ for atomic programs

$$\bar{v}(\mathtt{H}(i)) = I^{\otimes i} \otimes H \otimes I^{\otimes n-i-1}, \quad \bar{v}(\mathtt{X}(i)) = I^{\otimes i} \otimes X \otimes I^{\otimes n-i-1},$$
$$\bar{v}(\mathtt{Y}(i)) = I^{\otimes i} \otimes Y \otimes I^{\otimes n-i-1}, \quad \bar{v}(\mathtt{Z}(i)) = I^{\otimes i} \otimes Z \otimes I^{\otimes n-i-1},$$
$$\bar{v}(\mathtt{CX}(i,j)) = I^{\otimes i} \otimes |0\rangle\langle 0| \otimes I^{\otimes n-i-1} + (I^{\otimes i} \otimes |1\rangle\langle 1| \otimes I^{\otimes n-i-1})(I^{\otimes j} \otimes X \otimes I^{\otimes n-j-1}),$$
$$\bar{v}(\mathtt{SWAP}(i,j)) = \bar{v}(\mathtt{CX}(i,j) \,;\, \mathtt{CX}(j,i) \,;\, \mathtt{CX}(i,j)),$$

where $I^{\otimes i} = \overbrace{I \otimes \cdots \otimes I}^{i}$.

Standard Interpretation

- Standard interpretation $\overline{V} : L_0 \to \mathcal{C}(\mathbb{C}^{2^n})$ for atomic formulas

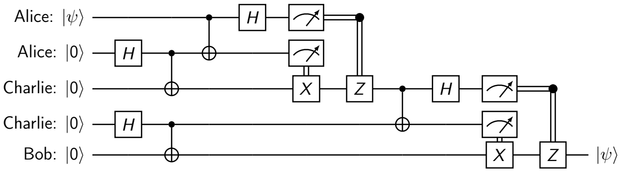
$$\overline{V}(p(i, |\psi\rangle)) = \mathbb{C}^{2^i} \otimes \mathrm{span}\{|\psi\rangle\} \otimes \mathbb{C}^{2^{n-i-1}},$$
$$\overline{V}(p(i, i+1, |\Psi\rangle)) = \mathbb{C}^{2^i} \otimes \mathrm{span}\{|\Psi\rangle\} \otimes \mathbb{C}^{2^{n-i-2}},$$

- Conditional quantum programs for quantum tests with probability in PDQL:

$$\text{if } \mathrm{P}^{\geq r} A \text{ then } a \text{ else } b \text{ fi} = (A?^{\geq r} \,;\, a) \cup (\neg A?^{\leq (1-r)} \,;\, b)$$

☞ considering binary projective measurements

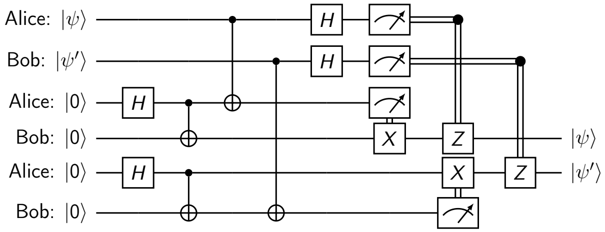
Quantum Relay Scheme



$$
\begin{aligned}
\text{relay} = {} & \texttt{H}(1)\ ;\ \texttt{CX}(1,2)\ ;\ \texttt{H}(3)\ ;\ \texttt{CX}(3,4)\ ;\ \texttt{CX}(0,1)\ ;\ \texttt{H}(0) \\
& ;\ \text{if } p(1, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{X}(2) \text{ fi} \\
& ;\ \text{if } p(0, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{Z}(2) \text{ fi} \\
& ;\ \texttt{CX}(2,3)\ ;\ \texttt{H}(2) \\
& ;\ \text{if } p(3, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{X}(4) \text{ fi} \\
& ;\ \text{if } p(2, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{Z}(4) \text{ fi}
\end{aligned}
$$

We verify that "a pure state $|\psi\rangle$ is correctly teleported" for Quantum Relay Scheme as follows:

$$(\overline{M}_5, |\psi\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle) \models [\text{relay}]p(4, |\psi\rangle)$$

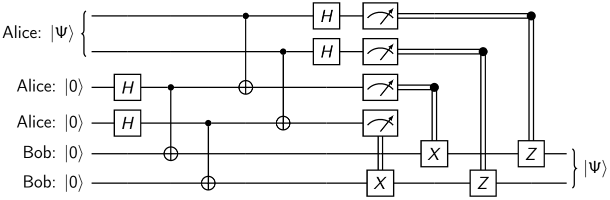


$$
\begin{aligned}
\text{biTeleport} = \; & \texttt{H}(2) \; ; \texttt{CX}(2,3) \; ; \texttt{H}(4) \; ; \texttt{CX}(4,5) \\
& ; \texttt{CX}(0,2) \; ; \texttt{CX}(1,5) \; ; \texttt{H}(0) \; ; \texttt{H}(1) \\
& ; \text{if } p(2, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{X}(3) \text{ fi} \\
& ; \text{if } p(0, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{Z}(3) \text{ fi} \\
& ; \text{if } p(5, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{X}(4) \text{ fi} \\
& ; \text{if } p(1, |0\rangle)^{\geq 1/2} \text{ then skip else } \texttt{Z}(4) \text{ fi}
\end{aligned}
$$

We verify that "two pure states $|\psi\rangle$ and $|\psi'\rangle$ owned by two users are correctly teleported to each other" for Bidirectional Quantum Teleportation as follows:

$$(\overline{M}_6, |\psi\rangle \otimes |\psi'\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle) \models [\text{biTeleport}] p(3, |\psi\rangle) \wedge p(4, |\psi'\rangle)$$

Two-qubit Quantum Teleportation



$$
\begin{aligned}
\text{twoTeleport} = {} & \mathtt{H}(2) \,;\, \mathtt{H}(3) \,;\, \mathtt{CX}(2,4) \,;\, \mathtt{CX}(3,5) \\
& ;\, \mathtt{CX}(0,2) \,;\, \mathtt{CX}(1,3) \,;\, \mathtt{H}(0) \,;\, \mathtt{H}(1) \\
& ;\, \text{if } p(3, |0\rangle)^{\geq 1/2} \text{ then skip else } \mathtt{X}(5) \text{ fi} \\
& ;\, \text{if } p(2, |0\rangle)^{\geq 1/2} \text{ then skip else } \mathtt{X}(4) \text{ fi} \\
& ;\, \text{if } p(1, |0\rangle)^{\geq 1/2} \text{ then skip else } \mathtt{Z}(5) \text{ fi} \\
& ;\, \text{if } p(0, |0\rangle)^{\geq 1/2} \text{ then skip else } \mathtt{Z}(4) \text{ fi}
\end{aligned}
$$

We verify that "arbitrary two-qubit pure states $|\Psi\rangle$ is correctly teleported" for Two-qubit Quantum Teleportation as follows:

$$(\overline{M}_6, |\Psi\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle) \models [\text{twoTeleport}]p(4, 5, |\Psi\rangle)$$

A Support Tool and Experiment Results

- A support tool for PDQL is extended from our previous support tool for BDQL[7] to handle the probabilistic operator $P^{\geq r}$.
- The implementation is available at `https://github.com/canhminhdo/DQL`

Protocol	Qubits	Rewrite Steps	Verification Time
Superdense Coding	2	2,451	1ms
Quantum Teleportation	3	9,034	4ms
Quantum Secret Sharing	4	39,041	18ms
Entanglement Swapping	4	14,272	6ms
Quantum Relay Scheme	5	44,939	26ms
Bidirectional Quantum Teleportation	6	47,717	27ms
Two-qubit Quantum Teleportation	6	660,313	238ms
Quantum Gate Teleportation	6	667,806	250ms
Quantum Network Coding	14	11,568,281	4,811ms

[7] Takagi, Do, and Ogata, "Automated Quantum Program Verification in a Dynamic Quantum Logic".

Contents

1 Introduction

2 Basic Notations on Quantum Computation

3 Dynamic Quantum Logic (DQL)

4 Probabilistic Dynamic Quantum Logic (PDQL)

5 Case Studies

6 Conclusions and Future Work

Conclusions and Future Work

- We have extended BDQL to PDQL by introducing the probabilistic operator $\mathsf{P}^{\geq r}$.
- A support tool has been developed in Maude to automate the formal verification of several well-known existing quantum programs.
- We consider several lines of future work as follows:
  - Conduct more case studies where the probabilistic properties are realistically expressed, such as Quantum Search Algorithm and Quantum Leader Election Protocol.
  - Handle properties related to iteration (quantum loop).
  - Extend PDQL to verify properties for concurrent quantum programs.

Thank You!