Termination and Universal Termination Problems for Nondeterministic Quantum Programs

MING XU, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
JIANLING FU, School of Computer Science and Technology, East China Normal University, China
HUI JIANG, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
YUXIN DENG, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China
ZHI-BIN LI, School of Computer Science and Technology, East China Normal University, China

Verifying quantum programs has attracted a lot of interest in recent years. In this paper, we consider the following two categories of termination problems of quantum programs with nondeterminism, namely:

(1) (termination) Is an input of a program terminating with probability one under all schedulers? If not, how can a scheduler be synthesized to evidence the nontermination?
(2) (universal termination) Are all inputs terminating with probability one under their respective schedulers? If yes, a further question asks whether there is a scheduler that forces all inputs to be terminating with probability one together with how to synthesize it; otherwise, how can an input be provided to refute the universal termination?

For the effective verification of the first category, we over-approximate the reachable set of quantum program states by the reachable subspace, whose algebraic structure is a linear space. On the other hand, we study the set of divergent states from which the program terminates with probability zero under *some* scheduler. The divergent set also has an explicit algebraic structure. Exploiting these explicit algebraic structures, we address the decision problem by a necessary and sufficient condition, i. e. the disjointness of the reachable subspace and the divergent set. Furthermore, the scheduler synthesis is completed in exponential time, whose bottleneck lies in computing the divergent set, reported for the first time.

For the second category, we reduce the decision problem to the existence of invariant subspace, from which the program terminates with probability zero under *all* schedulers. The invariant subspace is characterized by linear equations and thus can be efficiently computed. The states on that invariant subspace are evidence of the nontermination. Furthermore, the scheduler synthesis is completed by seeking a pattern of finite schedulers that forces all inputs to be terminating with positive probability. The repetition of that pattern yields the desired universal scheduler that forces all inputs to be terminating with probability one. All the problems in the second category are shown, also for the first time, to be solved in polynomial time. Finally, we demonstrate the aforementioned methods via a running example — the quantum Bernoulli factory protocol.

CCS Concepts: • **Theory of computation → Verification by model checking**; • **Software and its engineering → Formal software verification**; • **Security and privacy → Logic and verification**.

Additional Key Words and Phrases: quantum program, Markov decision process, termination, controller synthesis, fixedpoint

Authors' addresses: Ming Xu, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China, mxu@cs.ecnu.edu.cn; Jianling Fu, School of Computer Science and Technology, East China Normal University, Shanghai, China, scsse_fjl2015@126.com; Hui Jiang, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China, yhq_jh@126.com; Yuxin Deng, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China, yxdeng@sei.ecnu.edu.cn; Zhi-Bin Li, School of Computer Science and Technology, East China Normal University, Shanghai, China, lizb@cs.ecnu.edu.cn.

ACM 1557-7392/2024/9-ART
https://doi.org/10.1145/3691632

1 INTRODUCTION

In the field of quantum computing, physical devices have been rapidly developed in the last decades, particularly in very recent years. From the original experimental Deutsch's problem on a working 2-qubit quantum computer in 1998 [12], to in 2020 Chinese quantum computer JiuZhang's implementing a type of Boson sampling on 76 photons [74], and in 2023 IBM's releasing its latest 133-bit "Heron" processor [32], quantum computers have showed the utility [29] of requiring less computing time, requiring less power, or yielding more accurate results, against their classical counterparts.

To harness the power of quantum computers and exert quantum mechanics, a series of quantum programming languages have emerged, including the first practical quantum procedural language QCL [57], quantum guarded command language (qGCL) to program a "universal" quantum computer [58] with nondeterministic extension [75], and functional programming languages QFC [59] and QML [3] with high-level features. Nowadays, several quantum programming languages, e. g., Qiskit [31], Q# [60] and Cirq [26], have been proposed for real-world applications. They enable researchers to develop quantum algorithms and software conveniently and efficiently, and bring the new trend of quantum software engineering [2, 55].

However, the correctness of quantum programs as well as classical programs cannot be guaranteed without formal verification [45], which can serve as a formal foundation for quality assurance in software engineering. For classical programs, C. A. R. Hoare devised a formal logic, called Hoare Logic, where correctness formulas are expressed as triples comprising precondition, program, and postcondition. Hoare logic can be used to analyze the properties of a program, so that total correctness was decomposed into partial correctness plus termination [30]. A powerful approach to proving termination is to synthesize a *ranking function* [9] that maps program states into a well-founded domain, thus no infinite execution implies termination. E. W. Dijkstra presented the weakest precondition calculus [17] that is another approach to proving total correctness. P. Cousot and R. Cousot presented the abstract interpretation [14] for static analysis, which is particularly favorable for quantum programs since the dynamical execution suffers from noises in the NISQ era. All these methods lead to automatic verification of program correctness, and are extended for probabilistic programs and quantum ones particularly in recent years (to be commented in Subsection 1.1). Additionally, nondeterminism is a useful mechanics in software design, providing a way to describe the framework of a program without specifying implementation details. So it should been taken into consideration in the theory of programming [21, 40].

In the setting of nondeterministic quantum programs, the **termination** problem asks whether *an input state* of a program terminating with probability one under *all* schedulers; the **weak termination** problem asks whether *an input state* of a program terminating with probability one under *some* scheduler; the **universal termination** problem asks whether *all input states* are terminating with probability one under their respective schedulers. In this paper, we study the termination and the universal termination problems, aiming to give a series of positive results toward solving them, and left the weak termination problem as future work.

Expressiveness on Program Models. First of all, as operational semantics, two models of quantum Markov decision processes are considered with states in finite-dimensional Hilbert spaces. One treats the program locations of finitely many executable statements as the classical states, so that it is more straightforward as a high-level language to model practical scenarios. The other embraces only one program location which can be omitted, so that it provides a simpler structure as an assembly language for the ease of verification. Both are meant to interpret the operational semantics of nondeterministic quantum programs. We show that they are of the same expressiveness in terms of termination probability, and thus adopt the latter for ease in this work.

Deciding Termination with Complexity Analysis. To decide the termination, we investigate two critical concepts involved in reachable states of a program, namely the reachable space and divergent set. In general, the set of reachable states does not exhibit any explicit algebraic structure, which brings nontrivial hardness in verification. To overcome it, we give two definitions of reachable space that over-approximate the set of reachable states.

- The I-reachable space has the type of a subspace of the Hilbert space spanned by those vector representations of reachable states, as proposed in [47].
- The II-reachable space is spanned by those matrix representations of reachable states.

Both are computable in polynomial time, specified in terms of the size of program model as usual, i. e. the dimension of Hilbert space. But the latter is more precise, which is validated by the running example of quantum Bernoulli factory protocol. The divergent set consists of states from which the program terminates with probability zero under some scheduler. By exploiting the algebraic structure of the divergent set, an effective approach is developed to compute them in exponential time. Combining the reachable spaces and the divergent set, the termination is decided by a necessary and sufficient condition, i. e., the reachable subspace and the divergent set are disjoint. The complexity of the decision procedure is in exponential time, whose bottleneck lies in computing divergent set, which is reported for the first time.

Scheduler Synthesis. If the termination is decided to be false, we know there are some schedulers that force the program not to terminate with probability one. Scheduler synthesis plays an important role in resolving the nondeterminism, remaindered in the early software design. To achieve this, we confine the nontermination scheduler into finitely many ω-regular ones as candidates. Conditioning on each candidate, we derive a system of linear equations by Brouwer's fixedpoint theorem, whose nonzero solutions help us to identify the candidate as the nontermination scheduler.

Deciding Universal Termination. It is reduced to the existence of invariant subspace, from which the program terminates with probability zero under any scheduler. The invariant subspace is also characterized by linear equations and can be computed in polynomial time, thus deciding the universal termination. The states on that invariant subspace are evidences of the nontermination. If the universal termination is decided to be true, the scheduler synthesis is completed in polynomial time by seeking a pattern of finite schedulers that forces all input states to be terminating with positive probability. The repetition of that pattern yields the desired universal scheduler that forces all input states to be terminating with probability one.

1.1 Related Work

Verification on Probabilistic Programs. In contrast to deterministic programs, probabilistic programs have several new syntactic constructors — probabilistic choice, nondeterministic choice and observation [40], whose operational semantics are interpreted by Markov chains and Markov decision processes [5, Chapter 10] and supported by the verification tools PRISM [44] and Storm [15] much well.

Due to the aforementioned new syntactic constructors in probabilistic programs, the termination problem of probabilistic programs yields many variants to be studied, e. g.,

- *almost-sure termination* — Does a program terminate with probability one?
- *positive almost-sure termination* — Is the expected running time of a program finite?

Although the almost-sure termination of probabilistic programs was proved to be undecidable in general [37], there are many approaches to attacking it. Fioriti and Hermanns [24] proposed a framework to prove almost-sure termination by *ranking super-martingales*, which is analogous to ranking functions on deterministic programs. Chakarov and Sankaranarayanan [10] applied constraint-based techniques to generate linear ranking super-martingales. Chatterjee *et al.* [11] constructed polynomial ranking super-martingales extending linear ones. A polynomial-time procedure was given in [1] to synthesize lexicographic ranking super-martingales for linear probabilistic programs. Fu and Chatterjee [25] applied ranking super-martingales to study the positive almost-sure termination of probabilistic programs with nondeterministic choices, namely angelic and demonic choices.

A key step toward the automatic verification of probabilistic programs is how to define predicates over program states in a probabilistic distribution. There are two approaches to define probabilistic predicates. One [6] is to characterize the predicates of program states against program assertions by the membership relation of multiple sets; the other [42, 43, 53] is to adopt the expectations of the satisfaction degree. McIver and Morgan [50] generalized the *weakest preconditions* to the *weakest pre-expectations* for analyzing properties of probabilistic guarded command language (pGCL) [28] and for establishing almost-sure termination [51]. Kaminski *et al.* [38, 39] presented a calculus of weakest pre-expectation style for obtaining bounds on the expected running time of probabilistic programs. Verification tools like Amber [52] have been released to automatically prove almost-sure and positive almost-sure termination. The scheduler was synthesized to resolve nondeterminism in [8], such that the determinized probabilistic program meets a given quantitative specification including termination. However, in the setting of quantum computing, a program state is no longer simply a probabilistic distribution; it is instead a density operator (positive semi-definite matrix with unit trace) on Hilbert space, which would be considered in the following.

Verification on Quantum Programs. In 2004 Selinger [59] proposed the description of program states of a quantum program by density operators; In 2006 D'Hondt and Panangaden [16] adopted Hermitian operators (positive semi-definite matrices) as quantum predicates, which was further employed by [19] and [67] in reasoning about the correctness of deterministic quantum programs. Recently, Fang and Ying [18] described quantum states of quantum programs by stabilizers that are matrices stable/unchanged under the mapping of their defining operators; Guan *et al.* [27] used subspaces of the whole Hilbert space as quantum predicate for checking quantitative properties. Various formalism flourished the verification on quantum programs.

Ying and Feng [68] first studied the verification of quantum loop programs by giving some necessary and sufficient conditions to ensure termination and almost-sure termination. Quantum Markov chains [23, 27, 62, 63] could be a standard model to interpret the operational semantics of deterministic quantum programs. Yu and Ying [73] considered concurrent quantum programs, and reduced the termination problem to the reachability problem of quantum Markov chains. Extending quantum Markov chains with nondeterministic actions, quantum Markov decision processes [71] are suitable to interpret the operational semantics of nondeterministic quantum programs as in the present paper.

The work close to ours is [47] where Li *et al.* dealt with nondeterministic quantum programs. The nondeterminism in that program is used to model quantum processes, and the program execution relies on a scheduler to be specified by the software designer. Given a nondeterministic quantum program, the set of reachable states from an input state has no explicit algebraic structure in general, which yields nontrivial hardness in verification. The authors of [47] proposed a polynomial-time method for computing a linear space named by the reachable space, over-approximating the reachable set. They also presented an algorithm to compute the set of divergent states but the time complexity of the algorithm was left unsettled. When the two sets are disjoint, the termination of a program can be inferred. However, two remaining issues should be addressed, i. e., i) how to analyze the complexity of computing the divergent set and ii) how to synthesize the scheduler for nontermination. Both will be solved in the present work.

The classical Hoare logic was extended in the quantum setting to quantum Hoare logic [64] supporting both classical and quantum variables [22], and the Sharir–Pnueli–Hart method was also extended from probabilistic programs to quantum programs [70] toward automatic verification [66]. Ying [69] proposed the notions of additive and multiplicative invariants that are predicates over program states at program locations for quantum programs with angelic and demonic choices. Inspired by the aforementioned ranking super-martingale for probabilistic programs [11, 25], linear ranking super-martingales were defined and further synthesized in [46] w. r. t. an additive and/or multiplicative invariant by semi-definite programming and quantifier elimination over real closed fields. It gave a sound approach to the termination problems, since the existence of linear ranking super-martingales is a sufficient but not necessary condition to termination.

D'Hondt and Panangaden [16] proposed a quantum analogy of the weakest preconditions by using Hermitian operators as quantum predicate. Quantum weakest precondition was designed for reasoning about the expected runtime of quantum program [49] and the expected cost of various quantum resources [4]. The hardness of synthesizing it lies in the fact that the precondition is no longer linear on the superposition of pure states, which is admitted in quantum mechanics. To solve it, an approach [49] was presented based on quantum Hoare logic [48, 64]. For the state space admitting a finite-dimensional Hilbert space, it was also revealed in [49] that almost-sure termination of a deterministic quantum program with an input state is equivalent to positive almost-sure termination. The finite dimension in state space could bring extra sugar in our analysis, breaking the obstacle of undecidability for probabilistic programs [37].

Recently, Yu and Palsberg [72] presented a framework of quantum abstract interpretation to check assertions for the properties of large-scale quantum programs. The relationship between abstract interpretation and Hoare logic was also revealed in [20].

1.2 Contribution and Innovation

The contributions of the current paper are summarized as follows:

(1) We propose a precise over-approximation of the reachable set, which can be computed in polynomial time.
(2) The complexity of computing the set of divergent states is given for the first time, thus settling an open problem originally posed in [47].
(3) We decide the termination problem in exponential time and synthesize a nontermination scheduler provided that it exists.
(4) We decide the universal termination problem in polynomial time together with the synthesis of a universal scheduler for termination.

To achieve them, our technical innovations lie in: i) using a tree construction for demonstrating the derivation of divergent states, whose explicit structure could analyze the complexity of deciding termination; ii) applying Knaster–Tarski fixedpoint theorem on *discrete* lattice structures to ensure the efficiency of computing reachable spaces while applying Brouwer's fixedpoint theorem on *continuous* state spaces to ensure the effectivity of scheduler synthesis.

Organization. The rest of this paper is organized as follows. Section 2 recalls basic notions and notations from quantum computing. The models of nondeterministic quantum programs are introduced in Section 3 together with their termination problems. We compute the reachable spaces and the divergent set respectively in Sections 4 & 5. Combining them, we are able to decide the termination in Section 6. We further solve the universal termination problem in Section 7. Section 8 is the conclusion. For clarity, the implementation details is moved to the appendix.

2 PRELIMINARIES

Let $\mathbb{H}$ be a finite Hilbert space that is a complete vector space over complex numbers $\mathbb{C}$ equipped with an inner product operation, and d the dimension of $\mathbb{H}$ throughout this paper. We recall the standard Dirac notations from quantum computing. Interested readers can refer to [56] for more details.

- $|\psi\rangle$ stands for a unit column vector in $\mathbb{H}$ labelled with ψ;
- $\langle\psi| := |\psi\rangle^\dagger$ is the Hermitian adjoint (transpose and complex conjugate entrywise) of $|\psi\rangle$;
- $\langle\psi_1|\psi_2\rangle := \langle\psi_1|\,|\psi_2\rangle$ is the inner product of $|\psi_1\rangle$ and $|\psi_2\rangle$;
- $|\psi_1\rangle\langle\psi_2| := |\psi_1\rangle \otimes \langle\psi_2|$ is the outer product where $\otimes$ denotes tensor product;
- $|\psi, \psi'\rangle$ is a shorthand of the tensor product $|\psi\rangle\,|\psi'\rangle = |\psi\rangle \otimes |\psi'\rangle$.

Let $\{|i\rangle : i = 1, 2, \ldots, d\}$ be an orthonormal basis of $\mathbb{H}$. Then any element $|\psi\rangle$ of $\mathbb{H}$, interpreted as a *state*, can be entirely determined as $|\psi\rangle = \sum_{i=1}^{d} c_i\,|i\rangle$, where $c_i \in \mathbb{C}$ $(i = 1, 2, \ldots, d)$ satisfy the normalization condition $\sum_{i=1}^{d} |c_i|^2 = 1$. If $|\psi\rangle$ is linearly expressed by two or more elements $|i\rangle$ with nonzero coefficients, it is said to be a *super-position* of those elements $|i\rangle$. For two spaces $\mathbb{S}$ and $\mathbb{S}'$, the join $\mathbb{S} \vee \mathbb{S}'$ is the space spanned by the elements of $\mathbb{S}$ and $\mathbb{S}'$, i. e. $\mathrm{span}(\mathbb{S} \cup \mathbb{S}')$. For two quantum systems $\mathbb{H}$ and $\mathbb{H}'$, the state space of their composite system is given by the tensor product $\mathbb{H} \otimes \mathbb{H}'$ that is the Hilbert space spanned by the tensor products of elements in $\mathbb{H}$ and $\mathbb{H}'$, i. e. $\mathrm{span}(\{|\psi, \psi'\rangle : |\psi\rangle \in \mathbb{H} \wedge |\psi'\rangle \in \mathbb{H}'\})$, equipped with the inner product $\langle \psi_1, \psi_1' \mid \psi_2, \psi_2' \rangle = \langle \psi_1 | \psi_2 \rangle \langle \psi_1' \mid \psi_2' \rangle$ for any $|\psi_1\rangle, |\psi_2\rangle \in \mathbb{H}$ and $|\psi_1'\rangle, |\psi_2'\rangle \in \mathbb{H}'$.

Linear Operator. Let γ be a linear operator on $\mathbb{H}$. It is *Hermitian* if $\gamma = \gamma^\dagger$. A Hermitian operator γ is entirely determined by its d diagonal elements $\langle i| \gamma |i\rangle \in \mathbb{R}$ $(i = 1, 2, \ldots, d)$ and $d(d-1)/2$ off-diagonal elements $\langle i| \gamma |j\rangle \in \mathbb{C}$ with $1 \leq i < j \leq d$ for a total of d^2 real numbers. Let $\mathcal{H}(\mathbb{H})$ be the set of Hermitian operators on $\mathbb{H}$. For brevity, such a parameter $\mathbb{H}$ in $\mathcal{H}(\mathbb{H})$ can be omitted if it is clear from the context. For a Hermitian operator γ, we have the spectral decomposition $\gamma = \sum_{i=1}^{d} \lambda_i |\lambda_i\rangle\langle\lambda_i|$ where $\lambda_i \in \mathbb{R}$ $(i = 1, 2, \ldots, d)$ are the eigenvalues of γ and $|\lambda_i\rangle$ (taking the meaningful labels λ_i) are the corresponding eigenvectors. The *support* of γ is the subspace contained in $\mathbb{H}$ spanned by all eigenvectors associated with nonzero eigenvalues, i. e. $\mathrm{supp}(\gamma) := \mathrm{span}(\{|\lambda_i\rangle : i = 1, 2, \ldots, d \wedge \lambda_i \neq 0\})$. Although the spectral decomposition of γ may be not unique, the support of γ must be unique, since it is the orthocomplement of the null space $\mathrm{span}(\{|\psi\rangle \in \mathbb{H} : \gamma |\psi\rangle = 0\})$ of γ. So the notion of support is well defined. A Hermitian operator γ is *positive* if $\langle \psi| \gamma |\psi\rangle \geq 0$ holds for any $|\psi\rangle \in \mathbb{H}$. A *projector* $\mathbf{P}$ is a positive operator of the form $\sum_{i=1}^{m} |\psi_i\rangle\langle\psi_i|$ with $m \leq d$, where $|\psi_i\rangle$ $(i = 1, 2, \ldots, m)$ are orthonormal. It implies that all eigenvalues of $\mathbf{P}$ are either 0 or 1.

Quantum State. The *trace* of a linear operator γ is defined as $\mathrm{tr}(\gamma) := \sum_{i=1}^{d} \langle \psi_i| \gamma |\psi_i\rangle$ for any orthonormal basis $\{|\psi_i\rangle : i = 1, 2, \ldots, d\}$. It is unique as it equals the sum of all eigenvalues of γ. A *density* operator ρ is a positive operator with unit trace; a partial density operator ρ is a positive operator with trace not greater than 1. Let $\mathcal{D}$ be the set of density operators, and $\mathcal{D}^{\leq 1}$ the set of partial density operators. For a density operator ρ, we have the spectral decomposition $\rho = \sum_{i=1}^{m} \lambda_i |\lambda_i\rangle\langle\lambda_i|$ where λ_i $(i = 1, 2, \ldots, m)$ are positive eigenvalues. We call such eigenvectors $|\lambda_i\rangle$ *eigenstates* of ρ explained below. The density operators are usually used to describe quantum states. Under that decomposition, it means that the quantum system is in state $|\lambda_i\rangle$ with probability λ_i. When $m = 1$, we know that the system is surely in state $|\lambda_1\rangle$ (with probability one), which is the so-called *pure* state; otherwise the state is *mixed*. Both the vector notation $|\lambda_i\rangle$ and the outer product notation $|\lambda_i\rangle\langle\lambda_i|$ could be employed to denote pure states. An alternative way to describe quantum states is the *probabilistic ensemble* $\{(|\psi_k\rangle, p_k) : k = 1, 2, \ldots\}$ with $\sum_k p_k = 1$. It means that the system is the mixture of being in state $|\psi_k\rangle$ with probability weight p_k. Here $|\psi_k\rangle$ $(k = 1, 2, \ldots)$ are not necessarily orthogonal.

Quantum Operation. Super-operators $\mathcal{E}$ on $\mathbb{H}$ are linear operators on the (ground) linear operators on $\mathbb{H}$. Particularly, (completely-positive) super-operators are employed to describe quantum operations. It is usually described by the Kraus representation $\mathcal{E} = \{\mathbf{E}_i : i = 1, 2, \ldots, m\}$, entailing that for a given density operator ρ, we have $\mathcal{E}(\rho) = \sum_{i=1}^{m} \mathbf{E}_i \rho \mathbf{E}_i^\dagger$. Here the number m of *Kraus operators* $\mathbf{E}_i$ could be bounded by d^2 without loss of generality, since the (ground) linear operators ρ are Hermitian, i. e. a linear space of dimension d^2, and thereby there are at most d^2 linearly independent linear operators on that space. We will use the bracket notation $\{\mathbf{E}_i : i = 1, 2, \ldots, m\}$ to denote (the Kraus representation of) a super-operator $\mathcal{E}$. For two super-operators $\mathcal{E} = \{\mathbf{E}_i : i = 1, 2, \ldots, m\}$ and $\mathcal{E}' = \{\mathbf{E}_j' : j = 1, 2, \ldots, m'\}$, their sum $\mathcal{E} + \mathcal{E}'$ is given by $\{\mathbf{E}_i : i = 1, 2, \ldots, m\} \cup \{\mathbf{E}_j' : j = 1, 2, \ldots, m'\}$; their composition $\mathcal{E} \circ \mathcal{E}'$ with associative law $\mathcal{E} \circ \mathcal{E}'(\rho) = \mathcal{E}(\mathcal{E}'(\rho))$ is $\{\mathbf{E}_i \mathbf{E}_j' : i = 1, 2, \ldots, m \wedge j = 1, 2, \ldots, m'\}$. Let $\mathcal{I}$ be the identity super-operator, and $\mathbf{I}$ the identity operator. A super-operator $\mathcal{E}$ is *trace-preserving*, denoted $\mathcal{E} \approx \mathcal{I}$, if $\sum_{i=1}^{m} \mathbf{E}_i^\dagger \mathbf{E}_i = \mathbf{I}$, due to

$$\mathrm{tr}(\mathcal{E}(\rho)) = \mathrm{tr}\left(\sum_{i=1}^{m} \mathbf{E}_i \rho \mathbf{E}_i^\dagger\right) = \mathrm{tr}\left(\sum_{i=1}^{m} \mathbf{E}_i^\dagger \mathbf{E}_i \rho\right) = \mathrm{tr}(\mathbf{I}\rho) = \mathrm{tr}(\rho);$$

it is *trace-nonincreasing*, denoted $\mathcal{E} \lesssim \mathcal{I}$, if $\mathbf{I} - \sum_{i=1}^m \mathbf{E}_i^\dagger \mathbf{E}_i$ is positive, due to

$$\begin{aligned}\mathrm{tr}(|\psi\rangle\langle\psi|) - \mathrm{tr}(\mathcal{E}(|\psi\rangle\langle\psi|)) &= \mathrm{tr}(\mathbf{I}\,|\psi\rangle\langle\psi|) - \mathrm{tr}\left(\sum_{i=1}^m \mathbf{E}_i^\dagger \mathbf{E}_i\,|\psi\rangle\langle\psi|\right)\\ &= \mathrm{tr}\left(\left(\mathbf{I} - \sum_{i=1}^m \mathbf{E}_i^\dagger \mathbf{E}_i\right)|\psi\rangle\langle\psi|\right) = \langle\psi|\left(\mathbf{I} - \sum_{i=1}^m \mathbf{E}_i^\dagger \mathbf{E}_i\right)|\psi\rangle \geq 0.\end{aligned}$$

Let $\mathcal{S}$ be the set of super-operators, $\mathcal{S}^{\eqsim\mathcal{I}}$ the set of trace-preserving ones, and $\mathcal{S}^{\lesssim\mathcal{I}}$ the set of trace-nonincreasing ones. Clearly, $\mathcal{E} \in \mathcal{S}^{\eqsim\mathcal{I}}$ means both $\mathcal{E} \in \mathcal{S}^{\lesssim\mathcal{I}}$ and $\mathcal{E} \in \mathcal{S}^{\gtrsim\mathcal{I}}$.

Quantum Measurement. A finite set of projectors $\mathbf{P}_i$ with index i ranging over *IDX* forms a *projective measurement* if $\sum_{i\in IDX} \mathbf{P}_i = \mathbf{I}$. The measurement aims to extract classical information from quantum states, but it may destroy the quantum state. Specifically, given a quantum state ρ, after the above projective measurement, we will get an outcome $i \in IDX$ with probability $p_i = \mathrm{tr}(\mathbf{P}_i\rho)$; when the outcome is i, the final state would be *collapsed* to $\mathbf{P}_i\rho\mathbf{P}_i/p_i$. The measurement process is not reversible. For a projector $\mathbf{P}_i$ and a quantum state ρ, $\mathrm{tr}(\mathbf{P}_i\rho) = 0$ implies that the outcome i does not occur, which holds if and only if $\mathrm{supp}(\mathbf{P}_i)$ is orthogonal to $\mathrm{supp}(\rho)$. For a super-operator $\mathcal{E} = \{\mathbf{E}_i : i = 1, 2, \ldots, m\}$ and a pure state $|\psi\rangle\langle\psi|$, we have $\mathrm{supp}(\mathcal{E}(|\psi\rangle\langle\psi|)) = \mathrm{span}(\{\mathbf{E}_i\,|\psi\rangle : i = 1, 2, \ldots, m\})$. Finally we would mention a useful inclusion:

$$\mathrm{supp}(\mathcal{E}(|\psi\rangle\langle\psi|)) \subseteq \bigvee_{k=1}^{K} \mathrm{supp}(\mathcal{E}(|\psi_k\rangle\langle\psi_k|)) \tag{1}$$

holds for any $|\psi\rangle \in \mathrm{span}(\{|\psi_k\rangle : k = 1, 2, \ldots, K\})$. It follows from the fact: Assume the RHS of (1) is a proper subspace contained in $\mathbb{H}$; otherwise the inclusion follows trivially. Let $|\psi^\perp\rangle$ be an element of $\mathbb{H}$ orthogonal to all $\mathrm{supp}(\mathcal{E}(|\psi_k\rangle\langle\psi_k|))$ with $k = 1, 2, \ldots, K$. It is also orthogonal to $\mathbf{E}_i\,|\psi_k\rangle$ for each $i = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, K$, i. e. $\langle\psi^\perp|\mathbf{E}_i\,|\psi_k\rangle = 0$. It implies that $\langle\psi^\perp|\mathbf{E}_i\,|\psi\rangle = 0$ holds for each $i = 1, 2, \ldots, m$, and thus $|\psi^\perp\rangle$ is orthogonal to $\mathrm{supp}(\mathcal{E}(|\psi\rangle\langle\psi|))$.

3 PROGRAM MODEL AND TERMINATION PROBLEMS

In this section, we introduce a nondeterministic extension of a quantum *while*-language, and interpret its operational semantics by two models of quantum Markov decision processes (quantum MDPs). One model is more complicated but easier to model practical scenarios while the other is simpler and easier to be verified. They are shown to have the same expressiveness. For ease of verification, we will adopt the latter to represent nondeterministic quantum programs. Finally, we propose the termination problems of nondeterministic quantum programs considered in the paper.

3.1 Program Model

Definition 3.1 ([65, Chapter 6]). A nondeterministic quantum program is generated by the following syntax:

$$\begin{aligned} S \ \triangleq\ & q := |0\rangle \mid \bar{q} := U[\bar{q}] \mid S_1; S_2 \mid \square_{j=1}^m S_j\\ & \mid \textbf{if } \mathbf{M}[\bar{q}] = \textbf{true then } S_1\\ & \mid \textbf{while } \mathbf{M}[\bar{q}] = \textbf{true do } S_1 \ \ . \end{aligned}$$

We briefly explain the syntax as follows:

- The initialization "$q := |0\rangle$" sets quantum variable q to the basis state $|0\rangle$ on $\mathbb{H}_q$, where the Hilbert space $\mathbb{H}_q$ of q is supposed to have basis states $|0\rangle_{\mathrm{bool}}, |1\rangle_{\mathrm{bool}}$ for q being Boolean and $|0\rangle_{\mathrm{int}}, |1\rangle_{\mathrm{int}}, \ldots, |k-1\rangle_{\mathrm{int}}$ for q being integer. Such subscripts 'bool' and 'int' can be omitted if they are clear from the context. For any appointed pure state $|\psi\rangle \in \mathbb{H}$, there is a unitary operation $\mathbf{U}$ such that the state $|\psi\rangle = \mathbf{U}|0\rangle$ can be prepared.

- The assignment "$\bar{q} := U[\bar{q}]$" performs the unitary transformation U on the register $\bar{q}$. For operations in classical programs, we can implement the quantum analogue by constructing a corresponding unitary operator. For example, we can take the unitary operation $\mathbf{U}_{+1} = \sum_{i=0}^{2^{32}-1} |i+1 \mod 2^{32}\rangle\langle i|$ as the quantum counterpart to implement the classical increment assignment $x := x + 1$ on a 32-bit integer x.
- The statement "$S_1; S_2$" represents the sequential composition.
- The nondeterministic choice "$\square_{j=1}^{m} S_j$" means that a subprogram S_j ($j = 1, 2, \ldots, m$) is nondeterministically chosen to execute. Without loss of generality, the number of nondeterministic choices is set as a constant m. Then the nondeterminism will be resolved by some action α_j taken from the set $\mathbf{ACT} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$, indicating the subprogram S_j to be executed.
- The statement

$$\textbf{if } \mathbf{M}[\bar{q}] = \textbf{true then } S_1$$

is a quantum analogue of the classical condition statement. As the guard condition, a quantum measurement $\mathbf{M} = \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\}$ is performed on the register $\bar{q}$. If the outcome of the measurement is **true** whose probability is $p_{\mathbf{true}} = \mathrm{tr}(\mathbf{M}_{\mathbf{true}}[\bar{q}])$, the state in register $\bar{q}$ is collapsed into $\bar{q}_{\mathbf{true}} := \mathbf{M}_{\mathbf{true}}[\bar{q}]/p_{\mathbf{true}}$, and the subprogram S_1 will be applied to $\bar{q}_{\mathbf{true}}$, resulting in $S_1[\bar{q}_{\mathbf{true}}]$; otherwise the state in register $\bar{q}$ is collapsed into $\bar{q}_{\mathbf{false}} := \mathbf{M}_{\mathbf{false}}[\bar{q}]/p_{\mathbf{false}}$ where $p_{\mathbf{false}} = \mathrm{tr}(\mathbf{M}_{\mathbf{false}}[\bar{q}])$ and the subprogram S_1 will not be applied. It is easy to see $p_{\mathbf{true}} + p_{\mathbf{false}} = 1$. Note that if $p_{\mathbf{true}} = 0$ (resp. $p_{\mathbf{false}} = 0$), meaning the outcome **true** (resp. **false**) cannot be observed, this branch is not necessary to be further considered. In the viewpoint of static analysis, the resulting state would be the mixture of $S_1[\bar{q}_{\mathbf{true}}]$ and $\bar{q}_{\mathbf{false}}$ with probability $p_{\mathbf{true}}$ and $p_{\mathbf{false}}$, respectively.
- The loop statement

$$\textbf{while } \mathbf{M}[\bar{q}] = \textbf{true do } S_1$$

admits a projective measurement $\mathbf{M} = \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\}$ as the guard condition, so that exactly one of the two outcome **true** and **false** would occur after performing that measurement. If the outcome **false** is observed, the program ends in the collapsed state $\bar{q}_{\mathbf{false}}$; if the outcome **true** occurs, the subprogram S_1 will be applied to $\bar{q}_{\mathbf{true}}$ and repeat the above process. The resulting state would be a mixture of countably many states in static analysis.

A nondeterministic quantum program is a *finite* statement generated by the syntax in Definition 3.1. For a program $\mathcal{P}$, we could assign each statement with a program location as specified in the left column of Table 1, in which the nondeterministic choice, the condition and the loop statements are of appropriate wrapping. All of such locations are collected into the finite set $\mathbf{LOC} = \{\ell_i : i = 1, 2, \ldots, n\}$, in which ℓ_n is the end location, indicating no statement to be executed. Let $\mathbf{VAR}$ be the set of program variables of $\mathcal{P}$, and $\mathbf{ACT}$ the set of actions that resolve the nondeterministic choices. For a variable $q \in \mathbf{VAR}$, the state space is $\mathbb{H}_q$. The state space $\mathbb{H}_{\mathbf{VAR}}$ of all program variables is simply the tensor product of individual state spaces $\mathbb{H}_q$, i. e. $\mathbb{H}_{\mathbf{VAR}} = \bigotimes_{q \in \mathbf{VAR}} \mathbb{H}_q$.

Example 3.2. We here consider a quantum Bernoulli factory protocol [35, 41] constructed on two quantum coins to serve as a running example of our method. Alice and Bob are meant to bet by applying the protocol, which in detail is described as:

(1) Two quantum coins, named *quoins*, are referred to as the left and the right ones.
(2) It nondeterministically chooses one of the two quoins to toss, and the other one is flipped.
(3) If the left quoin is head and the right is tail, then Alice wins; if the right quoin is head and the left is tail, then Bob wins; otherwise, they end in a draw.

Table 1. The operational semantics of nondeterministic quantum programs

statements	**operational semantics**
$\ell_i: \quad q := \|0\rangle$ $\ell_{i+1}: \quad S_1$	$(\ell_i, \rho) \xrightarrow{\tau} (\ell_{i+1}, \sum_{j=0}^{k-1} \|0\rangle_q\langle j\| \rho \|j\rangle_q\langle 0\|)$ where the subscript q indicates which variable is involved, and $k = \dim(\mathbb{H}_q)$
$\ell_i: \quad \bar{q} := U[\bar{q}]$ $\ell_{i+1}: \quad S_1$	$(\ell_i, \rho) \xrightarrow{\tau} (\ell_{i+1}, \mathbf{U}\rho\mathbf{U}^\dagger)$
$\ell_i: \quad \square_{j=1}^{m}$ $\ell_{i+j}: \quad S_j$ $\ell_{i+m+1}: \quad S_{m+1}$	$(\ell_i, \rho) \xrightarrow{\alpha_j} (\ell_{i+j}, \rho)$
$\ell_i: \quad$ **if** $\mathbf{M}[\bar{q}] =$ **true then** $\ell_{i+1}: \quad S_1$ $\ell_j: \quad S_2$	$(\ell_i, \rho) \xrightarrow{\tau} (\ell_{i+1}, \mathbf{M}_{\mathbf{true}}\rho\mathbf{M}_{\mathbf{true}})$ $(\ell_i, \rho) \xrightarrow{\tau} (\ell_j, \mathbf{M}_{\mathbf{false}}\rho\mathbf{M}_{\mathbf{false}})$
$\ell_i: \quad$ **while** $\mathbf{M}[\bar{q}] =$ **true do** $\ell_{i+1}: \quad S_1$ $\ell_j: \quad S_2$	$(\ell_i, \rho) \xrightarrow{\tau} (\ell_{i+1}, \mathbf{M}_{\mathbf{true}}\rho\mathbf{M}_{\mathbf{true}})$ $(\ell_i, \rho) \xrightarrow{\tau} (\ell_j, \mathbf{M}_{\mathbf{false}}\rho\mathbf{M}_{\mathbf{false}})$
$\ell_n:$	$(\ell_n, \rho) \xrightarrow{\tau} (\ell_n, \rho)$

Intuitively, tossing a quoin would produce the result "head" or "tail" with equal probability, independent to the initial status of the two quoins, so that it makes the bet fair. However, Alice and Bob want to know whether the result is *defective* in the sense that neither Alice nor Bob eventually has a chance of winning. Let us take Alice's stand to check the defectiveness in quantum setting, which is similar from Bob's stand.

In order to describe the process of checking the defectiveness, we design a nondeterministic quantum program $\mathcal{P}_1$ with program variables **VAR** $= \{q_1, q_2\}$ and locations **LOC** $= \{\ell_1, \ell_2, \ldots, \ell_9\}$ as follows.

```
$\ell_1$:  $q_1 := |0\rangle$
$\ell_2$:  $q_2 := |0\rangle$
$\ell_3$:  $q_1 := U_1[q_1]$
$\ell_4$:  $q_2 := U_2[q_2]$
$\ell_5$:  while $\mathbf{M}[q_1; q_2]$ = true do
    $\ell_6$:  $\square_{j=1}^{2}$
        $\ell_7$:  $(X_2 \circ H_1)[q_1; q_2]$
        $\ell_8$:  $(X_1 \circ H_2)[q_1; q_2]$
$\ell_9$:
```

Both $\mathbb{H}_{q_1}$ and $\mathbb{H}_{q_2}$ are the one-qubit Hilbert space with orthonormal basis $\{|0\rangle, |1\rangle\}$ where $|0\rangle$ and $|1\rangle$ denote "head" and "tail" respectively. The state space of the two program variables $\mathbb{H}_{\mathbf{VAR}} = \mathbb{H}_{q_1} \otimes \mathbb{H}_{q_2}$ is a two-qubit Hilbert space. The unitary transformations U_i ($i = 1, 2$) transform the initial one-qubit state $|0\rangle$ in registers q_i into any one-qubit state $|\psi\rangle$ to be prepared. For instance, we choose $U_i = \{\mathbf{X}\}$ where $\mathbf{X} = |0\rangle\langle 1| + |1\rangle\langle 0|$ is the bit-flip, so that $\mathbf{X}|0\rangle = |1\rangle$ is prepared in our setting. The status of two quoins prior to the *while*-loop can be viewed as the composite quantum state $q_1; q_2 := |1, 1\rangle \in \mathbb{H}_{\mathbf{VAR}}$.

For the while-loop, a projective measurement $\mathbf{M} = \{\mathbf{M_{true}}, \mathbf{M_{false}}\}$ is designed as the guard condition, where $\mathbf{M_{true}} = |0,1\rangle\langle 0,1|$ and $\mathbf{M_{false}} = \mathbf{I_{VAR}} - \mathbf{M_{true}} = |0,0\rangle\langle 0,0| + |1,0\rangle\langle 1,0| + |1,1\rangle\langle 1,1|$ are referred to the events "the left quoin is head and the right is tail" and the complement, respectively. Whenever we enter the while loop, i. e. being at location ℓ_6, a nondeterministic choice corresponding to tossing the left or the right quoin should be resolved by some action taken from the set $\mathbf{ACT} = \{\alpha_1, \alpha_2\}$, which leads to location ℓ_7 or ℓ_8 and the program will return to location ℓ_5 after that. Finally the program would be expected to terminate at location ℓ_9.

Tossing the quoin q_i is modelled by applying the Hadamard gate $H_i = \{\mathbf{H}\}$ where $\mathbf{H} = |+\rangle\langle 0| + |-\rangle\langle 1| = |0\rangle\langle +| + |1\rangle\langle -|$ with $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$ on the ith qubit, which means that $|0\rangle$ is transformed into $|+\rangle$ and $|1\rangle$ is transformed into $|-\rangle$, resulting in the super-positions of "head" and "tail" with equal probability. Flipping the quoin q_i is modelled by applying the bit-flip gate $X_i = \{\mathbf{X}\}$ on the ith qubit.

3.2 Operational Semantics

We have seen that during the execution of a program, the program states should take program locations into consideration. To this end, we will interpret the operational semantics of a nondeterministic quantum program $\mathcal{P}$ by a model of quantum MDP on the Hilbert space $\mathbb{H}_{\mathbf{VAR}}$ (quantum information) with program locations $\mathbf{LOC}$ (classical information).

Let us review the model of quantum MDP first.

Definition 3.3. A quantum Markov decision process (quantum MDP for short) on Hilbert space $\mathbb{H}$ is a quadruple $(S, \Sigma, \mathcal{E}, \mathbf{M})$, in which

- $S = \{s_i : i = 1, 2, \ldots, n\}$ is a finite set of classical states;
- $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$ is a finite set of actions;
- $\mathcal{E} : (S \times \Sigma \times S) \to \mathcal{S}^{\lesssim \mathcal{I}}$ gives rise to the super-operators $\mathcal{E}_{i,j,k}$ on $\mathbb{H}$ that characterize the transitions from state s_i to s_k by taking action α_j, satisfying that $\sum_{s_k \in S} \mathcal{E}_{i,j,k} \eqsim \mathcal{I}$ holds for each $s_i \in S$ and each $\alpha_j \in \Sigma$;
- $\mathbf{M}$ is a projective measurement on $\mathbb{H}_{\mathrm{cq}} = C \otimes \mathbb{H}$ with $C = \mathrm{span}(\{|s\rangle : s \in S\})$.

Note that in the classical model of MDP, there is a probability-allocation function attached to state transitions, which is generalized to the density operator-allocation function by those super-operators $\mathcal{E}_{i,j,k}$ in Definition 3.3. Additionally, to extract classical information from quantum states, the projective measurement $\mathbf{M}$ is adopted here.

Definition 3.4. For a nondeterministic quantum program $\mathcal{P}$ with program variables $\mathbf{VAR}$, actions $\mathbf{ACT}$ and locations $\mathbf{LOC}$, the quantum MDP that interprets $\mathcal{P}$ is a quadruple $(\mathbf{LOC}, \mathbf{ACT}, \to, \{\overline{\mathbf{M}}_{\mathbf{t}}, \overline{\mathbf{M}}_{\mathbf{nt}}\})$ on Hilbert space $\mathbb{H}_{\mathbf{VAR}}$, where

- the transition relation $\to$, whose entries $(\ell_i, \rho) \xrightarrow{\alpha_j} (\ell_k, \rho')$ characterize the nondeterministic transitions from location ℓ_i to ℓ_k by taking action α_j with changing quantum states ρ to ρ' while $(\ell_i, \rho) \xrightarrow{\tau} (\ell_k, \rho')$ characterize the deterministic transitions with the wildcard τ of actions in $\mathbf{ACT}$, is given by the quantum operations in the right column of Table 1, and
- $\{\overline{\mathbf{M}}_{\mathbf{t}}, \overline{\mathbf{M}}_{\mathbf{nt}}\}$ is a projective measurement on $\mathrm{span}(\{|\ell\rangle : \ell \in \mathbf{LOC}\})$, in which $\overline{\mathbf{M}}_{\mathbf{t}} = |\ell_n\rangle\langle \ell_n|$ refers to the end location of $\mathcal{P}$ and $\overline{\mathbf{M}}_{\mathbf{nt}} = \sum_{\ell \in \mathbf{LOC} \setminus \{\ell_n\}} |\ell\rangle\langle \ell|$ refers to the complement.

Here, the projectors $\overline{\mathbf{M}}_{\mathbf{t}}$ and $\overline{\mathbf{M}}_{\mathbf{nt}}$ on the space $\mathrm{span}(\{|\ell\rangle : \ell \in \mathbf{LOC}\})$ are short for the trivial extensions $\overline{\mathbf{M}}_{\mathbf{t}} \otimes \mathbf{I_{VAR}}$ and $\overline{\mathbf{M}}_{\mathbf{nt}} \otimes \mathbf{I_{VAR}}$ on the product Hilbert space $\mathrm{span}(\{|\ell\rangle : \ell \in \mathbf{LOC}\}) \otimes \mathbb{H}_{\mathbf{VAR}}$. The transitional super-operator function $\mathcal{E}$ in quantum MDP is defined on two parameters — the locations and the actions. When interpreting the deterministic statements, the action $\alpha \in \mathbf{ACT}$ could not be specified and the transition relations keep the same whenever the wildcard τ is thought of an arbitrary action.

When a nondeterministic program $\mathcal{P}$ executes, it has finitely many actions $\alpha_1, \alpha_2, \ldots, \alpha_m$ to choose at each location ℓ_i of nondeterministic statements. Each action $\alpha_j \in \mathbf{ACT}$ is attached with a series of super-operators

$\mathcal{E}_{i,j,k}$, where ℓ_k ranges over **LOC**, satisfying $\sum_{\ell_k \in \mathbf{LOC}} \mathcal{E}_{i,j,k} \eqsim \mathcal{I}$. The nondeterminism is resolved by a scheduler, i. e., a sequence of actions. An infinite sequence $\sigma = \sigma(1)\,\sigma(2)\cdots$ with $\sigma(i) \in \mathbf{ACT}$ is called an *infinite scheduler* (scheduler for short), and a finite sequence $\varsigma = \sigma(1)\,\sigma(2)\cdots\sigma(k)$ is a *finite scheduler.*

Example 3.5. Consider the nondeterministic quantum program $\mathcal{P}_1$ with actions $\mathbf{ACT} = \{\alpha_1, \alpha_2\}$ and locations $\mathbf{LOC} = \{\ell_1, \ell_2, \dots, \ell_9\}$ in Example 3.2. Since the program terminates at the location ℓ_9, we can obtain a projective measurement $\{\overline{\mathbf{M}}_\mathbf{t}, \overline{\mathbf{M}}_\mathbf{nt}\}$ with $\overline{\mathbf{M}}_\mathbf{t} = |\ell_9\rangle\langle\ell_9|$ and $\overline{\mathbf{M}}_\mathbf{nt} = \sum_{\ell \in \mathbf{LOC}\setminus\{\ell_9\}} |\ell\rangle\langle\ell|$. Thus we construct a quantum MDP $\mathcal{M}_1$ interpreting $\mathcal{P}_1$, which is characterized by the quadruple $(\mathbf{LOC}, \mathbf{ACT}, \rightarrow, \{\overline{\mathbf{M}}_\mathbf{t}, \overline{\mathbf{M}}_\mathbf{nt}\})$ with the transition relation $\rightarrow$ given by the middle column of Table 2. Here $\rho_1 = (\mathbf{I} \otimes \langle 0|)\rho(\mathbf{I} \otimes |0\rangle) + (\mathbf{I} \otimes \langle 1|)\rho(\mathbf{I} \otimes |1\rangle)$ and $\rho_2 = (\langle 0| \otimes \mathbf{I})\rho(|0\rangle \otimes \mathbf{I}) + (\langle 1| \otimes \mathbf{I})\rho(|1\rangle \otimes \mathbf{I})$ are the reduced density operators of ρ that trace out the states on $\mathbb{H}_{q_2}$ and $\mathbb{H}_{q_1}$, respectively.

Table 2. Translating the nondeterministic quantum program to the quantum MDPs

original statements in the quantum program $\mathcal{P}_1$ from Example 3.2	**transition relation $\rightarrow$ in the quantum MDP $\mathcal{M}_1$ as described in Definition 3.4**	**transition super-operator $\mathcal{E}$ in the quantum MDP $\mathcal{M}_1'$ as described in Definition 3.6**
$\ell_1:\quad q_1 := \lvert 0\rangle$ $\ell_2:\quad \cdots$	$(\ell_1, \rho) \xrightarrow{\tau} (\ell_2, \lvert 0\rangle\langle 0\rvert \otimes \rho_2)$	add Kraus operators $\lvert\ell_2\rangle\langle\ell_1\rvert \otimes \lvert 0\rangle\langle 0\rvert \otimes \mathbf{I}$ and $\lvert\ell_2\rangle\langle\ell_1\rvert \otimes \lvert 0\rangle\langle 1\rvert \otimes \mathbf{I}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_2:\quad q_2 := \lvert 0\rangle$ $\ell_3:\quad \cdots$	$(\ell_2, \rho) \xrightarrow{\tau} (\ell_3, \rho_1 \otimes \lvert 0\rangle\langle 0\rvert)$	add Kraus operators $\lvert\ell_3\rangle\langle\ell_2\rvert \otimes \mathbf{I} \otimes \lvert 0\rangle\langle 0\rvert$ and $\lvert\ell_3\rangle\langle\ell_2\rvert \otimes \mathbf{I} \otimes \lvert 0\rangle\langle 1\rvert$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_3:\quad q_1 := U_1[q_1]$ $\ell_4:\quad \cdots$	$(\ell_3, \rho) \xrightarrow{\tau} (\ell_4, U_1(\rho))$	add Kraus operator $\lvert\ell_4\rangle\langle\ell_3\rvert \otimes \mathbf{X} \otimes \mathbf{I}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_4:\quad q_2 := U_2[q_2]$ $\ell_5:\quad \cdots$	$(\ell_4, \rho) \xrightarrow{\tau} (\ell_5, U_2(\rho))$	add Kraus operator $\lvert\ell_5\rangle\langle\ell_4\rvert \otimes \mathbf{I} \otimes \mathbf{X}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_5:\quad$ **while** $\mathbf{M}[q_1; q_2] = \mathbf{true}$ **do** $\quad\ell_6:\quad \cdots$ $\ell_9:\quad \cdots$	$(\ell_5, \rho) \xrightarrow{\tau} (\ell_6, \mathbf{M}_\mathbf{true}\rho\mathbf{M}_\mathbf{true})$ $(\ell_5, \rho) \xrightarrow{\tau} (\ell_9, \mathbf{M}_\mathbf{false}\rho\mathbf{M}_\mathbf{false})$	add Kraus operators $\lvert\ell_6\rangle\langle\ell_5\rvert \otimes \lvert 0,1\rangle\langle 0,1\rvert$ and $\lvert\ell_9\rangle\langle\ell_5\rvert \otimes (\lvert 0,0\rangle\langle 0,0\rvert + \lvert 1,0\rangle\langle 1,0\rvert + \lvert 1,1\rangle\langle 1,1\rvert)$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_6:\quad \square_{j=1}^{2}$ $\quad\ell_7:\quad \cdots$ $\quad\ell_8:\quad \cdots$	$(\ell_6, \rho) \xrightarrow{\alpha_1} (\ell_7, \rho)$ $(\ell_6, \rho) \xrightarrow{\alpha_2} (\ell_8, \rho)$	add Kraus operator $\lvert\ell_7\rangle\langle\ell_6\rvert \otimes \mathbf{I}_\mathbf{VAR}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$ with $\vec{\alpha}(6) = \alpha_1$, add Kraus operator $\lvert\ell_8\rangle\langle\ell_6\rvert \otimes \mathbf{I}_\mathbf{VAR}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$ with $\vec{\alpha}(6) = \alpha_2$
$\ell_5:\quad$ **while** $\cdots$ **do** $\quad\ell_7:\quad (X_2 \circ H_1)[q_1; q_2]$	$(\ell_7, \rho) \xrightarrow{\tau} (\ell_5, (X_2 \circ H_1)(\rho))$	add Kraus operator $\lvert\ell_5\rangle\langle\ell_7\rvert \otimes \mathbf{H} \otimes \mathbf{X}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
$\ell_5:\quad$ **while** $\cdots$ **do** $\quad\ell_8:\quad (X_1 \circ H_2)[q_1; q_2]$	$(\ell_8, \rho) \xrightarrow{\tau} (\ell_5, (X_1 \circ H_2)(\rho))$	add Kraus operator $\lvert\ell_5\rangle\langle\ell_8\rvert \otimes \mathbf{X} \otimes \mathbf{H}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$
ℓ_9:	$(\ell_9, \rho) \xrightarrow{\tau} (\ell_9, \rho)$	add Kraus operator $\lvert\ell_9\rangle\langle\ell_9\rvert \otimes \mathbf{I}_\mathbf{VAR}$ to $\mathcal{E}(\vec{\alpha})$ for all $\vec{\alpha} \in \Sigma$

Starting at the location ℓ_1 and given a finite action sequence $\varsigma = \alpha_1\,\alpha_1\,\alpha_1\,\alpha_1\,\alpha_1\,\alpha_2\,\alpha_1$, the run of the quantum MDP $\mathcal{M}_1$ generated by ς is

$$\begin{aligned}(l_1, \rho) &\xrightarrow{\alpha_1} (\ell_2, |0\rangle\langle 0| \otimes \rho_2) \xrightarrow{\alpha_1} (\ell_3, |0,0\rangle\langle 0,0|) \xrightarrow{\alpha_1} (\ell_4, |1,0\rangle\langle 1,0|) \xrightarrow{\alpha_1} (\ell_5, |1,1\rangle\langle 1,1|)\\ &\xrightarrow{\alpha_1} (\ell_6, |1,1\rangle\langle 1,1|) \xrightarrow{\alpha_2} (\ell_8, |1,1\rangle\langle 1,1|) \xrightarrow{\alpha_1} (\ell_5, |0,-\rangle\langle 0,-|).\end{aligned}$$

The quantum state would evolve into $|0,-\rangle\langle 0,-|$ while $\mathcal{M}_1$ runs into the location ℓ_5. □

Sometimes, we would focus on the structure of *while*-loop that plays a central role in termination analysis. It is a subclass of nondeterministic quantum programs, which terminates when refuting the guard condition instead of entering the end location. Thus the location information can be omitted for brevity. We could interpret the while-loop by the model of quantum MDP as follows:

Definition 3.6 ([47, Definition 1]). For a nondeterministic quantum while-loop $\mathcal{P}$ with program variables **VAR** and actions **ACT**, the quantum MDP that interprets $\mathcal{P}$ is a triple $(\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ on Hilbert space $\mathbb{H}_{\mathbf{VAR}}$, where:

- $\Sigma = \mathbf{ACT}$;
- $\mathcal{E} : \Sigma \to \mathcal{S}^{\approx\mathcal{I}}$ gives rise to the super-operators $\mathcal{E}_j$ on $\mathbb{H}_{\mathbf{VAR}}$ by taking action α_j;
- $\{\mathbf{M_{true}}, \mathbf{M_{false}}\}$ is a projective measurement on $\mathbb{H}_{\mathbf{VAR}}$ with the outcomes **true** and **false** referring to the nontermination and the termination, respectively.

It is worth noting that in this model, a measurement is performed on the current quantum state to determine whether the program has already terminated before taking each action. In case the program does not terminate, an action α_j will be nondeterministically chosen and the corresponding super-operator $\mathcal{E}_j$ will be applied to the current quantum state. The program keeps running step by step like the above execution until it terminates, but it is unnecessary to consider the change on the location after executing every step.

Example 3.7. Review the nondeterministic quantum program $\mathcal{P}_1$ in Example 3.2. There is an embedded quantum while-loop $\mathcal{P}_2$ (from location ℓ_5 to ℓ_9) with guard condition $\{\mathbf{M_{true}}, \mathbf{M_{false}}\}$. We can interpret it simply by the quantum MDP $\mathcal{M}_2 = (\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ with $\Sigma = \{\alpha_1, \alpha_2\}$ and the input state $\rho_0 = |1,1\rangle\langle 1,1|$, where

$$\begin{aligned}\mathcal{E}(\alpha_1) &= \mathcal{E}_1 = X_2 \circ H_1 = \{\mathbf{H} \otimes \mathbf{X}\}\\ \mathcal{E}(\alpha_2) &= \mathcal{E}_2 = X_1 \circ H_2 = \{\mathbf{X} \otimes \mathbf{H}\}.\end{aligned}$$

We define the Kraus operators $\mathbf{E}_1 = \mathbf{H} \otimes \mathbf{X}$ and $\mathbf{E}_2 = \mathbf{X} \otimes \mathbf{H}$ on $\mathbb{H}_{\mathbf{VAR}}$ for use afterwards. □

Although the model in Definition 3.4 seems much easier to manipulate than that in Definition 3.6, they are of the same expressiveness indicated by the following lemma. Here, the same expressiveness means that for each scheduler σ_1 in the model in Definition 3.6, there exists a corresponding scheduler σ_2 in the model in Definition 3.4 such that two models embrace the same termination probability, and vice versa. Hence, we can freely choose one of the two definitions for convenience. By this result, we indicate that nondeterministic quantum while-loops are generic to describe nondeterministic quantum programs, and thus we will mainly adopt the model in Definition 3.6 for ease of verification.

Lemma 3.8. *The model in Definition 3.4 has the same expressiveness as that in Definition 3.6.*

Proof. Given a quantum MDP $(\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ on Hilbert space $\mathbb{H}_{\mathbf{VAR}}$ in Definition 3.6, we can obtain another quantum MDP $(\mathbf{LOC}, \mathbf{ACT}, \to, \{\overline{\mathbf{M}}_\mathbf{t}, \overline{\mathbf{M}}_\mathbf{nt}\})$ on Hilbert space $\mathbb{H}_{\mathbf{VAR}}$, as described in Definition 3.4, by introducing two locations ℓ_1 and ℓ_2. Then every transition of $\mathcal{E}$ makes a self-loop at ℓ_1 if it is not terminating, otherwise is led to ℓ_2. Formally it is constructed as:

- setting the location set $\mathbf{LOC} = \{\ell_1, \ell_2\}$, so that $\overline{\mathbf{M}}_\mathbf{t} = |\ell_2\rangle\langle\ell_2|$ and $\overline{\mathbf{M}}_\mathbf{nt} = |\ell_1\rangle\langle\ell_1|$,

- setting the action set $\mathbf{ACT} = \Sigma$,
- $(\ell_1, \rho) \xrightarrow{\alpha_j} (\ell_2, \mathbf{M_{false}} \rho \mathbf{M_{false}})$ and $(\ell_1, \rho) \xrightarrow{\alpha_j} (\ell_1, \rho')$ follow from $\rho' = \mathcal{E}_j(\mathbf{M_{true}} \rho \mathbf{M_{true}})$ where $\mathcal{E}_j = \mathcal{E}(\alpha_j)$ for $\alpha_j \in \Sigma$.

Conversely, given a quantum MDP $(\mathbf{LOC}, \mathbf{ACT}, \rightarrow, \{\overline{\mathbf{M}}_{\mathbf{t}}, \overline{\mathbf{M}}_{\mathbf{nt}}\})$ with locations $\mathbf{LOC} = \{\ell_1, \ell_2, \dots, \ell_n\}$ on Hilbert space $\mathbb{H}_{\mathbf{VAR}}$ in Definition 3.4, we can obtain another quantum MDP $(\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ on the product Hilbert space $C \otimes \mathbb{H}_{\mathbf{VAR}}$ with $C = \mathrm{span}(\{|\ell\rangle : \ell \in \mathbf{LOC})$, as described in Definition 3.6, by quantitizing the location information $\mathbf{LOC}$ into C. Formally it is constructed as:

- setting the action set $\Sigma = \mathbf{ACT}^n$ that is the n-fold of $\mathbf{ACT}$, whose elements are n-tuples with ith component specifying the action at location ℓ_i as in the original quantum MDP $(\mathbf{LOC}, \mathbf{ACT}, \rightarrow, \{\overline{\mathbf{M}}_{\mathbf{t}}, \overline{\mathbf{M}}_{\mathbf{nt}}\})$,
- $\mathcal{E}(\vec{\alpha})(\sum_{i=1}^n |\ell_i\rangle\langle\ell_i| \otimes \rho_i) = \sum_{i=1}^n \sum_{\ell_{k_i} \in \mathbf{LOC}} |\ell_{k_i}\rangle\langle\ell_{k_i}| \otimes \rho_{k_i}$, where $\mathcal{E}(\vec{\alpha})$ is the super-operator of $\mathcal{E}$ by taking an action $\vec{\alpha} = (\alpha_{j_1}, \alpha_{j_2}, \dots, \alpha_{j_n}) \in \Sigma$, follows from the series of $(\ell_i, \rho_i) \xrightarrow{\alpha_{j_i}} (\ell_{k_i}, \rho_{k_i})$ with ℓ_{k_i} ranging over $\mathbf{LOC}$,
- setting $\mathbf{M_{false}} = |\ell_n\rangle\langle\ell_n| \otimes \mathbf{I_{VAR}}$ and $\mathbf{M_{true}} = \sum_{\ell \in \mathbf{LOC} \setminus \{\ell_n\}} |\ell\rangle\langle\ell| \otimes \mathbf{I_{VAR}}$.

From the above mutual construction, either model can simulate the other, which derives the same expressiveness of two models. □

Example 3.9. For the quantum MDP $\mathcal{M}_1 = (\mathbf{LOC}, \mathbf{ACT}, \rightarrow, \{\overline{\mathbf{M}}_{\mathbf{t}}, \overline{\mathbf{M}}_{\mathbf{nt}}\})$ in Example 3.5, we can construct an equally expressive quantum MDP $\mathcal{M}_1' = (\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ with the following components:

- the input state $|\ell_1\rangle\langle\ell_1| \otimes \rho_0$ for the input state ρ_0 of $\mathcal{M}_1$,
- the action set $\Sigma = \mathbf{ACT}^9$ as $|\mathbf{LOC}| = 9$,
- the transition super-operator $\mathcal{E}$ is constructed part by part in the right column of Table 2 and summarized as
$$\mathcal{E}(\vec{\alpha}) = \left\{ \begin{array}{l} |\ell_2\rangle\langle\ell_1| \otimes |0\rangle\langle 0| \otimes \mathbf{I},\ |\ell_2\rangle\langle\ell_1| \otimes |0\rangle\langle 1| \otimes \mathbf{I}, \\ |\ell_3\rangle\langle\ell_2| \otimes \mathbf{I} \otimes |0\rangle\langle 0|,\ |\ell_3\rangle\langle\ell_2| \otimes \mathbf{I} \otimes |0\rangle\langle 1|, \\ |\ell_4\rangle\langle\ell_3| \otimes \mathbf{X} \otimes \mathbf{I},\ |\ell_5\rangle\langle\ell_4| \otimes \mathbf{I} \otimes \mathbf{X}, \\ |\ell_6\rangle\langle\ell_5| \otimes |0,1\rangle\langle 0,1|,\ |\ell_9\rangle\langle\ell_5| \otimes (|0,0\rangle\langle 0,0| + |1,0\rangle\langle 1,0| + |1,1\rangle\langle 1,1|), \\ \underline{|\ell_7\rangle\langle\ell_6| \otimes \mathbf{I_{VAR}}},\ |\ell_5\rangle\langle\ell_7| \otimes \mathbf{H} \otimes \mathbf{X},\ |\ell_5\rangle\langle\ell_8| \otimes \mathbf{X} \otimes \mathbf{H},\ |\ell_9\rangle\langle\ell_9| \otimes \mathbf{I_{VAR}} \end{array} \right\}$$
if the 6th component $\vec{\alpha}(6)$ of the 9-tuple $\vec{\alpha} \in \Sigma$ is α_1, while replacing the underlined Kraus operator $|\ell_7\rangle\langle\ell_6| \otimes \mathbf{I_{VAR}}$ with $|\ell_8\rangle\langle\ell_6| \otimes \mathbf{I_{VAR}}$ if $\vec{\alpha}(6) = \alpha_2$, and
- the projectors $\mathbf{M_{false}} = |\ell_9\rangle\langle\ell_9| \otimes \mathbf{I_{VAR}}$ and $\mathbf{M_{true}} = \sum_{\ell \in \mathbf{LOC} \setminus \{\ell_9\}} |\ell\rangle\langle\ell| \otimes \mathbf{I_{VAR}}$. □

An execution scheduler of a program can be represented as a sequence of actions as in Definition 3.6. We define the super-operator $\mathcal{F}_{\alpha_j} = \mathcal{E}_j \circ \{\mathbf{M_{true}}\}$ $(\alpha_j \in \Sigma)$ as the composite quantum operation upon the measure outcome of nontermination; let $\varsigma \uparrow k$ be the finite prefix of ς with length k for $k \le |\varsigma|$, and $\varsigma \downarrow k$ the suffix obtained by removing the k-prefix from ς. Then we have the inductive construction of the super-operator over a sequence of actions
$$\mathcal{F}_\varsigma = \begin{cases} \mathcal{I} & \text{if } |\varsigma| = 0 \\ \mathcal{F}_{\varsigma\downarrow 1} \circ \mathcal{F}_{\varsigma\uparrow 1} & \text{if } |\varsigma| \ge 1. \end{cases}$$
For example, for a finite schedule $\varsigma = \alpha_1\alpha_2\alpha_3$, we have $\varsigma \uparrow 1 = \alpha_1$, $\varsigma \downarrow 1 = \alpha_2\alpha_3$, and $\mathcal{F}_\varsigma = \mathcal{F}_{\alpha_1\alpha_2\alpha_3} = \mathcal{F}_{\alpha_2\alpha_3} \circ \mathcal{F}_{\alpha_1} = \mathcal{F}_{\varsigma\downarrow 1} \circ \mathcal{F}_{\varsigma\uparrow 1}$. The construction of the super-operator over a sequence of actions can be extended to infinite schedulers σ.

From now on, we employ the model of quantum MDP $(\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ in Definition 3.6 to represent nondeterministic quantum programs $\mathcal{P}$. The size of $\mathcal{P}$ is dominated by $\mathcal{O}(m \cdot d^4)$ where $m = |\Sigma|$ and $d = \dim(\mathbb{H})$, since $\mathcal{E}$ has m super-operators $\mathcal{E}_j$ for $\alpha_j \in \Sigma$ and each super-operator $\mathcal{E}_j$ has at most d^2 Kraus operators that are d-by-d matrices. For brevity, we measure the size of $\mathcal{P}$ simply by the two parameters m and d. All the $m \cdot d^4$ numbers in $\mathcal{E}$ are supposed to be *algebraic numbers* that are roots of the polynomials with rational coefficients. Algebraic numbers are widely used in quantum computing, such as $1/\sqrt{2}$ appearing in the Hadamard gate and the imaginary unit i appearing in the Pauli gate $\mathrm{i}\,|1\rangle\langle 0| - \mathrm{i}\,|0\rangle\langle 1|$. Arithmetic operations (addition, subtraction, multiplication and division) on algebraic numbers are further supposed to be of unit cost as usual, i. e. $\mathcal{O}(1)$. These basics will build up our complexity analysis on later.

3.3 Termination Problems

We are to deliver the termination probabilities of nondeterministic quantum programs and the termination problems.

Definition 3.10 (Termination Probability). For a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ in Definition 3.6 and an input state $\rho \in \mathcal{D}$,

(1) the (accumulative) termination probability under a finite scheduler ς is

$$\mathrm{TP}_{\varsigma}(\rho) = \sum_{i=0}^{|\varsigma|} \mathrm{tr}(\mathbf{M}_{\mathbf{false}} \mathcal{F}_{\varsigma\uparrow i}(\rho));$$

(2) the termination probability under an infinite scheduler σ is

$$\mathrm{TP}_{\sigma}(\rho) = \sum_{i=0}^{\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{false}} \mathcal{F}_{\sigma\uparrow i}(\rho));$$

(3) the termination probability (for conservation) of $\mathcal{P}$ is $\mathrm{TP}(\rho) = \inf_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho)$.

It is not hard to see $\mathrm{TP}_{\varsigma}(\rho) = \mathrm{tr}(\rho) - \mathrm{tr}(\mathbf{M}_{\mathbf{true}} \mathcal{F}_{\varsigma}(\rho))$.

Problem 3.11 (Termination). Given a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ and an input state $\rho \in \mathcal{D}$, is ρ terminating with probability one under all schedulers σ, i. e. $\forall\, \sigma \in \Sigma^{\omega} : \mathrm{TP}_{\sigma}(\rho) = 1$? If not, how can a scheduler σ be synthesized to evidence the nontermination?

Problem 3.12 (Weak Termination). Given a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ and an input state $\rho \in \mathcal{D}$, is ρ terminating with probability one under some scheduler σ, i. e. $\exists\, \sigma \in \Sigma^{\omega} : \mathrm{TP}_{\sigma}(\rho) = 1$?

Problem 3.13 (Optimal Termination). Given a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ and an input state $\rho \in \mathcal{D}$, what is the angelic (resp. demonic) scheduler σ that maximizes (resp. minimizes) the termination probability, i. e. $\arg\max_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho)$ (resp. $\arg\min_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho)$)?

Problem 3.14 (Universal Termination). Given a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$, are all input states ρ terminating with probability one under their respective schedulers σ, i. e.

$$\forall\, \rho \in \mathcal{D}\ \exists\, \sigma \in \Sigma^{\omega} : \mathrm{TP}_{\sigma}(\rho) = 1? \tag{2a}$$

If yes, a further question asks whether there is a scheduler σ that forces all input states ρ to be terminating with probability one, i. e.

$$\exists\, \sigma \in \Sigma^{\omega}\ \forall\, \rho \in \mathcal{D} : \mathrm{TP}_{\sigma}(\rho) = 1, \tag{2b}$$

together with how to synthesize such a universal scheduler σ; otherwise, how can an input ρ be provided to refute the universal termination?

The first three problems are specified with an input state, while the last one is not, since it concerns the termination on all input states that is a "universal" problem. Specifically, Problem 3.11 requires the (strong) termination under *all* schedulers, Problem 3.12 requires the weak termination under *some* scheduler, both are concerned with *qualitative* properties. Problem 3.13 is on *quantitative* property, which seems to be harder than Problems 3.11 & 3.12, since for a given input state ρ, the program terminates with probability one under all schedulers if $\mathrm{TP}(\rho) = \inf_{\sigma\in\Sigma^\omega} \mathrm{TP}_\sigma(\rho) = 1$ while it terminates under some scheduler if $\sup_{\sigma\in\Sigma^\omega} \mathrm{TP}_\sigma(\rho) = 1$. However, as shown in [71, Example 1], such an optimal scheduler does not exist, thus Problem 3.13 is not well-posed sometimes. We will solve Problems 3.11 & 3.14 in the coming sections, and left the remaining Problem 3.12 as future work.

4 COMPUTING THE REACHABLE SPACES

In this section, we introduce the reachable spaces for a nondeterministic quantum program starting from an input state. They over-approximate the set of reachable states in order to obtain an explicit algebraic structure, which is crucial for an algorithmic analysis. We review the notion of reachable space together with the construction method presented in [47]. Then we propose a more precise notion of reachable space. Two kinds of reachable spaces are said to be of types I and II respectively, and both are computable in polynomial time w. r. t. the dimension of the state space $\mathbb{H}$ and the number of actions in **ACT** as the existing literature [47].

Definition 4.1 (Reachable Set). Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho \in \mathcal{D}$, the set of reachable states of $\mathcal{P}$ starting from ρ is $\Psi(\mathcal{P},\rho) = \{\mathcal{F}_\varsigma(\rho) : \varsigma \in \Sigma^*\}$.

The elements γ in the set $\Psi(\mathcal{P},\rho)$ are certainly reachable states from ρ. Here, the reachability is specified in a *qualitative* sense that requires a unit probability of reachability under some finite scheduler ς. Can we specify the reachability in a *quantitative* sense? To answer it, we investigate what states are in a given state $\rho \in \mathcal{D}$. Supposing that ρ is the uniform distribution $\mathbf{I}/d$, we know that:

- any pure state $|\psi\rangle \in \mathbb{H}$ is in ρ, which is with probability $\frac{1}{d}$,
- any state $\gamma \in \mathcal{D}$ is also in ρ, which is with probability $1/(d \cdot \lambda_{\max})$ where $\lambda_{\max}$ is the maximal eigenvalue of γ.

Overall, a state is in ρ, provided that it has a positive probability in some probabilistic ensemble of ρ. Developing this concept, a state is called *reachable* from ρ, provided that it has a positive probability of reachability under some finite scheduler ς. So the eigenstates $|\lambda\rangle$ with positive eigenvalue λ of $\gamma \in \Psi(\mathcal{P},\rho)$ are (pure) reachable states; and even the elements $|\psi\rangle$ in the support of γ are (pure) reachable states too, since, by [56, Exercise 2.73] (refer to Appendix A for self-containedness), there is a minimal probabilistic ensemble of γ containing $|\psi\rangle$ with positive probability p, i. e. $\gamma = p\,|\psi\rangle\langle\psi| + \sum_k p_k\,|\psi_k\rangle\langle\psi_k|$ for some $|\psi_k\rangle \in \mathrm{supp}(\gamma)$ with the probability sum $p + \sum_k p_k = 1$.

It is obvious to see that $\Psi(\mathcal{P},\rho)$ is a countable set without explicit algebraic structure in general, which yields nontrivial hardness in verification. To overcome it, we would like to introduce the notion of *reachable space*.

Definition 4.2 (I-Reachable Space [47, Definition 3]). Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho \in \mathcal{D}$, the type I reachable space of $\mathcal{P}$ starting from ρ is

$$\Phi(\mathcal{P},\rho) = \bigvee_{\gamma\in\Psi(\mathcal{P},\rho)} \mathrm{supp}(\gamma).$$

From the above definition, we can see that for two elements γ_1 and γ_2 of $\Psi(\mathcal{P},\rho)$ that are reachable under finite schedulers ς_1 and ς_2 respectively and for $|\psi_i\rangle \in \mathrm{supp}(\gamma_i)$ $(i \in \{1,2\})$, all super-positions $|\psi\rangle$ of $|\psi_1\rangle$ and $|\psi_2\rangle$ are elements of $\Phi(\mathcal{P},\rho)$, but they are unnecessarily required to be reachable since the construction does not guarantee the existence of a common finite scheduler ς that generates $|\psi\rangle$. In this sense, the I-reachable space is known to be a superset of the reachable set. More precisely, we have:

- $\Psi(\mathcal{P},\rho) \subset \mathcal{D}(\mathbb{H})$ since $\Psi(\mathcal{P},\rho)$ is countable while $\mathcal{D}(\mathbb{H})$ is a continuum that is uncountable,
- $\Phi(\mathcal{P},\rho) \subseteq \mathbb{H}$, and further

- $\Psi(\mathcal{P},\rho) \subset \mathcal{D}(\Phi(\mathcal{P},\rho))$.

Thus, to show that a property holds on the reachable set $\Psi(\mathcal{P},\rho)$, it is sufficient to show that the property holds on all density operators $\mathcal{D}(\Phi(\mathcal{P},\rho))$ on the reachable space $\Phi(\mathcal{P},\rho)$. The latter has the nice algebraic structure of a finite-dimensional linear space, which is promising to be effectively verified.

To get an explicit description of the reachable space, we resort to the following program model that has only one action and thus resolves nondeterminism:

Definition 4.3 (Average Quantum Program [47, Definition 4]). Let $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$ and $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ be a nondeterministic quantum program. Then the average quantum program $\bar{\mathcal{P}}$ of $\mathcal{P}$ is the pair $(\bar{\mathcal{E}}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$, where

- $\bar{\mathcal{E}}$ is the arithmetic average of $\mathcal{E}$, i. e., for any program state $\rho \in \mathcal{D}$, the effect of the average super-operator $\bar{\mathcal{E}}$ performed on ρ is $\frac{1}{m}\sum_{j=1}^{m}\mathcal{E}_j(\rho)$.

Lemma 4.4 ([47, Lemma 1]). *Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho \in \mathcal{D}$, the I-reachable subspace of $\mathcal{P}$ starting from ρ is that of the quantum program $\bar{\mathcal{P}}$ averaging $\mathcal{P}$ starting from ρ, i. e. $\Phi(\mathcal{P},\rho) = \Phi(\bar{\mathcal{P}},\rho)$.*

This lemma reveals that $\mathcal{P}$ agrees with $\bar{\mathcal{P}}$ on the I-reachable subspace, despite $\mathcal{P}$ does not on the reachable set. Using it, the I-reachable space of $\mathcal{P}$ can be obtained as the least fixedpoint of the ascending chain of linear subspaces of $\mathbb{H}$:

$$\begin{aligned} \operatorname{supp}(\varrho_0) &\subseteq \operatorname{supp}(\varrho_0) \vee \operatorname{supp}(\varrho_1) \\ &\subseteq \operatorname{supp}(\varrho_0) \vee \operatorname{supp}(\varrho_1) \vee \operatorname{supp}(\varrho_2) \\ &\subseteq \cdots, \end{aligned} \tag{3}$$

where $\varrho_i = \bar{\mathcal{F}}^i(\rho)$ with $\bar{\mathcal{F}} = \bar{\mathcal{E}} \circ \{\mathbf{M}_{\mathbf{true}}\}$. Namely, we denote this chain by $\mathbb{S}_0 \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_2 \subseteq \cdots$, in which each linear space $\mathbb{S}_i$ is computed upon the average quantum program $\bar{\mathcal{P}}$. The following lemma gives an upper bound for the occurrence of the least fixedpoint in the ascending chain, thus establishing the computability. Actually, it is embedded in the complexity analysis of [47, Algorithm 1]. To be self-contained, we deliver it as an independent result with proof.

Lemma 4.5. *Let $\mathbb{S}_0 \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_2 \subseteq \cdots$ be the ascending chain of nonnull linear subspaces $\mathbb{S}_i \subseteq \mathbb{H}$, as defined in* (3)*. Then there is an integer $\ell \leq \dim(\mathbb{H}) - 1$ such that $\mathbb{S}_k = \mathbb{S}_\ell$ holds for all $k > \ell$.*

Proof. The function F mapping from $\mathbb{S}_i$ to $\mathbb{S}_{i+1}$ $(i \geq 0)$ can be formulated as a monotonic function

$$F(\mathbb{X}) = \mathbb{X} \vee \bigvee_{|\psi\rangle \in \mathbb{X}} \operatorname{supp}(\bar{\mathcal{F}}(|\psi\rangle\langle\psi|)).$$

Meanwhile, all subspaces $\mathbb{S}^{\mathrm{all}}$ of $\mathbb{H}$ form a complete lattice $(\mathbb{S}^{\mathrm{all}}, \subseteq, \inf, \sup)$ by taking 'inf' as the meet $\bigwedge = \bigcap$ and 'sup' as the join $\bigvee$. By Knaster–Tarski fixedpoint theorem [14, 54], we have that the least fixedpoint occurs upon $\mathbb{S}_\ell = \mathbb{S}_{\ell+1}$, which ℓ is bounded by $\dim(\mathbb{H}) - 1$ since $\mathbb{S}_i$ are nonnull subspaces of $\mathbb{H}$. □

The procedure of computing the I-reachable space $\Phi(\mathcal{P},\rho)$ is stated in Algorithm 1 with complexity analysis below.

Complexity. The Kraus representation of $\mathcal{E}_j$ are known as the input information of $\mathcal{P}$. For convenience, we do not compute the simplest Kraus representation of $\bar{\mathcal{F}}$ whose number of Kraus operators can be bounded by d^2 here, but just use the average Kraus operators of $\mathcal{E}_j$, since the simplest Kraus representation obtained by quantum process tomography [56, Subsection 8.4.2] costs additionally $O(d^{12})$ operations. Note that there are less than d times of entering the inner loop in Line 6. Each inner loop performs l times of matrix-vector multiplication

Algorithm 1 Computing the I-Reachable Space [47, Algorithm 1]

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$ and $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ over $\mathbb{H}$ with dimension d, and an input state $\rho \in \mathcal{D}$;

Output: an orthonormal basis B of $\Phi(\mathcal{P}, \rho)$.

```
let $\bar{\mathcal{F}} = \frac{1}{m}\sum_{j=1}^{m} \mathcal{E}_j \circ \{\mathbf{M}_{\mathbf{true}}\}$ be the average super-operator;
let $\{\mathbf{F}_j : j = 1, 2, \ldots, l\}$ be a Kraus representation of $\bar{\mathcal{F}}$;
compute an orthonormal basis $B_0$ of supp$(\rho)$, and $B_{-1} \longleftarrow \emptyset$;
for $i \longleftarrow 1$ to $d - 1$ do
    $B_i \longleftarrow B_{i-1}$;
    for all $|\psi\rangle \in B_{i-1} \setminus B_{i-2}$ do
        $V \longleftarrow \{\mathbf{F}_j |\psi\rangle : j = 1, 2, \ldots, l\}$;
        compute an orthonormal basis $B'$ of $V$ complement to $B_i$;
        $B_i \longleftarrow B_i \cup B'$;
    if $B_i = B_{i-1}$ or $|B_i| = d$ then break;
return $B_i$.
```

and l times of computing orthocomplement, where l is bounded by $m \cdot d^2$, as the factor m comes from the number of actions in $\mathcal{P}$ and the factor d^2 comes from the number of Kraus operators of the super-operators $\mathcal{E}_j$. The matrix-vector multiplication $\mathbf{F}_j |\psi\rangle$ is in $O(d^2)$, and computing orthocomplement of $\mathbf{F}_j |\psi\rangle$ is also in $O(d^2)$ by normalizing the difference

$$\mathbf{F}_j |\psi\rangle - \sum_{|\varphi\rangle \in B_i} \langle\varphi| \mathbf{F}_j |\psi\rangle |\varphi\rangle$$

(if it is nonzero) as the standard Gram–Schmit procedure. Hence Algorithm 1 is in time $O(m \cdot d^5)$. □

Example 4.6. Continue to consider the nondeterministic quantum program $\mathcal{P}_1$ in Example 3.7, the average super-operator is $\bar{\mathcal{F}} = \frac{1}{2}(\mathcal{F}_{\alpha_1} + \mathcal{F}_{\alpha_2})$. Since $\frac{1}{2}\mathcal{F}_{\alpha_i}(\rho) = \frac{1}{2}(\mathbf{E}_i\mathbf{M}_{\mathbf{true}})\rho(\mathbf{E}_i\mathbf{M}_{\mathbf{true}})^\dagger$ for $i \in \{1, 2\}$, the Kraus representation of $\bar{\mathcal{F}}$ can be $\{\mathbf{F}_1, \mathbf{F}_2\}$, where

$$\mathbf{F}_1 = \frac{1}{\sqrt{2}}\mathbf{E}_1\mathbf{M}_{\mathbf{true}} = \frac{1}{\sqrt{2}}(|+, 1\rangle\langle 0, 0| + |-, 1\rangle\langle 1, 0| + |-, 0\rangle\langle 1, 1|),$$
$$\mathbf{F}_2 = \frac{1}{\sqrt{2}}\mathbf{E}_2\mathbf{M}_{\mathbf{true}} = \frac{1}{\sqrt{2}}(|1, +\rangle\langle 0, 0| + |0, +\rangle\langle 1, 0| + |0, -\rangle\langle 1, 1|).$$

By Algorithm 1, for the given input state $\rho_0 = |q_1, q_2\rangle\langle q_1, q_2| = |1, 1\rangle\langle 1, 1|$, the I-reachable space can be inductively computed as follows.

(1) Initially, we have $\mathbb{S}_0 = \text{supp}(\rho_0) = \text{span}(\{|1, 1\rangle\})$.

(2) To get the next subspace $\mathbb{S}_1$ along the ascending chain, for the basis element $|1, 1\rangle$ in $\mathbb{S}_0$, we compute

$$\mathbf{F}_1 |1, 1\rangle = \frac{1}{\sqrt{2}} |-, 0\rangle,$$
$$\mathbf{F}_2 |1, 1\rangle = \frac{1}{\sqrt{2}} |0, -\rangle.$$

The former operator $\mathbf{F}_1 |1, 1\rangle$ is already orthogonal to $\mathbb{S}_0$ and can be normalized to $|-, 0\rangle$; the latter operator $\mathbf{F}_2 |1, 1\rangle$ is also orthogonal to $\mathbb{S}_0$ but gives another orthogonal element $(|+, 0\rangle - \sqrt{2}|0, 1\rangle)/\sqrt{3}$ by normalizing $\frac{1}{\sqrt{2}}|0, -\rangle - \frac{1}{\sqrt{2}}\langle -, 0|0, -\rangle |-, 0\rangle$. Thus the orthonormal basis complement to $\mathbb{S}_0$ is $\{|-, 0\rangle, (|+, 0\rangle - \sqrt{2}|0, 1\rangle)/\sqrt{3}\}$, and we get $\mathbb{S}_1 = \text{span}(\{|1, 1\rangle, |-, 0\rangle, (|+, 0\rangle - \sqrt{2}|0, 1\rangle)/\sqrt{3}\})$.

(3) To get the subspace $\mathbb{S}_2$, for the newly-produced basis elements $|-,0\rangle$ and $(|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}$ in $\mathbb{S}_1$, we have

$$\begin{aligned} \mathbf{F}_1\,|-,0\rangle &= \tfrac{1}{\sqrt{2}}\,|1,1\rangle\,, \\ \mathbf{F}_2\,|-,0\rangle &= -\tfrac{1}{2}\,|-,+\rangle\,, \\ \mathbf{F}_1\tfrac{1}{\sqrt{3}}(|+,0\rangle - \sqrt{2}\,|0,1\rangle) &= \tfrac{1}{\sqrt{6}}\,|0,1\rangle\,, \\ \mathbf{F}_2\tfrac{1}{\sqrt{3}}(|+,0\rangle - \sqrt{2}\,|0,1\rangle) &= \tfrac{1}{\sqrt{6}}\,|+,+\rangle\,. \end{aligned}$$

Thus an orthonormal basis complement to $\mathbb{S}_1$ is $\{(-\sqrt{2}\,|+,0\rangle - |0,1\rangle)/\sqrt{3}\}$, and we get $\mathbb{S}_2 = \operatorname{span}(\{|1,1\rangle\,, |-,0\rangle\,, (|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}, (-\sqrt{2}\,|+,0\rangle - |0,1\rangle)/\sqrt{3}\})$. By $\dim(\mathbb{H}_{\mathbf{VAR}}) = d = 4 = \dim(\mathbb{S}_2)$ we have $\mathbb{S}_2 = \mathbb{H}_{\mathbf{VAR}}$.

Hence the least fixedpoint of the ascending chain occurs, which yields the I-reachable space $\Phi(\mathcal{P}_1, \rho_0) = \mathbb{H}_{\mathbf{VAR}}$. □

In the following, we will have a deeper observation of the reachable set and the reachable space. Since the former is a countable set and the latter is a continuum, the latter is possibly a much larger superset of the former. We are to narrow the over-approximation of the reachable set using other algebraic structures, instead of the I-reachable space. One promising way is to use the linearly independent basis of Hermitian operators on $\mathbb{H}$, say

$$\begin{aligned} &\{|i\rangle\langle i| : 1 \le i \le d\} \cup \{(|i\rangle\langle j| + |j\rangle\langle i|)/\sqrt{2} : 1 \le i < j \le d\} \\ &\qquad\qquad \cup \{(\mathrm{i}\,|i\rangle\langle j| - \mathrm{i}\,|j\rangle\langle i|)/\sqrt{2} : 1 \le i < j \le d\}. \end{aligned} \tag{4}$$

(When the Hilbert space $\mathbb{H}$ in consideration is exactly on the k-qubit system, i. e. $d = 2^k$ for some integer k, an alternative way is to use the 4^k linearly independent elements $\bigotimes_{i=1}^{k} \gamma_i$, where γ_i is one of Pauli operators $\mathbf{I}$, $\mathbf{X} = |0\rangle\langle 1| + |1\rangle\langle 0|$, $\mathbf{Y} = \mathrm{i}\,|0\rangle\langle 1| - \mathrm{i}\,|0\rangle\langle 1|$ and $\mathbf{Z} = |0\rangle\langle 0| - |1\rangle\langle 1|$.) Although the general state is expressed by all d^2 basis elements in (4), all reachable states might be expressed by only a part of these basis elements. So, using as few as possible basis elements to express all pure reachable states yields a more precise notion of reachable space. In the setting of reachability analysis, at most d^2 pure reachable states could be served as the linearly independent basis of $\mathcal{H}(\mathbb{H})$ that we require. To this end, we resort to the following operator-level program that characterizes the operations between pure reachable states.

Definition 4.7 (Operator-level Program). Let $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ be a nondeterministic quantum program with $\mathcal{E}_j = \{\mathbf{E}_{j,k} : k = 1, 2, \ldots, K_j\}$. Then the operator-level program $\hat{\mathcal{P}}$ of $\mathcal{P}$ is the triple $(\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$, where

- $\hat{\Sigma} = \{\alpha_{j,k} : j = 1, 2, \ldots, m \wedge k = 1, 2, \ldots, K_j\}$ is a finite set of actions;
- $\mathbf{E} : \hat{\Sigma} \longrightarrow \mathcal{L}$ gives rise to the linear operators $\mathbf{E}_{j,k}$ taken action $\alpha_{j,k}$, which are obtained from the Kraus representation $\bigcup_{j=1}^{m}\{\mathbf{E}_{j,k} : k = 1, 2, \ldots, K_j\}$ of $\sum_{j=1}^{m} \mathcal{E}_j$.

Rigorously speaking, the operator-level program $\hat{\mathcal{P}}$ is not a nondeterministic quantum program described in Definition 3.6, since the quantum operations do not meet the trace-preserving restriction generally, i. e., $\{\mathbf{E}_{j,k}\} \not\approx \mathcal{I}$ holds for all actions $\alpha_{j,k} \in \hat{\Sigma}$ where $\{\mathbf{E}_{j,k}\}$ denotes the super-operator that has the unique Kraus operator $\mathbf{E}_{j,k}$. However, dropping this restriction does not affect the qualitative termination $\mathrm{TP}(\rho) = 1$ considered in the paper, and we would study the qualitative termination of the operator-level program afterwards. For convenience, the notation $\mathbf{F}_\varsigma$ is adapted to $\mathcal{F}_\varsigma$, e. g. $\mathbf{F}_{\alpha_{j,k}} = \mathbf{E}_{j,k}\mathbf{M}_{\mathbf{true}}$ and $\mathbf{F}_\varsigma = \mathbf{F}_{\varsigma\downarrow 1}\mathbf{F}_{\varsigma\uparrow 1}$.

Definition 4.8 (II-Reachable Space). Given a nondeterministic quantum program $\mathcal{P}$ and an input pure state $\rho = |\lambda\rangle\langle\lambda| \in \mathcal{D}$, the type II reachable space of $\mathcal{P}$ starting from ρ is $\Upsilon(\mathcal{P}, \rho) = \operatorname{span}(\Psi(\hat{\mathcal{P}}, \rho))$, where $\hat{\mathcal{P}}$ is the operator-level program of $\mathcal{P}$ as in Definition 4.7.

It is not hard to see that the reachable set $\Psi(\mathcal{P}, \rho)$ is over-approximated by the II-reachable space $\Upsilon(\mathcal{P}, \rho)$, since i) all elements $\gamma \in \Psi(\mathcal{P}, \rho)$ can be linearly expressed by those elements in $\Psi(\hat{\mathcal{P}}, \rho)$ and ii) $\Upsilon(\mathcal{P}, \rho) = \mathrm{span}(\Psi(\hat{\mathcal{P}}, \rho))$.

For an input pure state $\rho = |\lambda\rangle\langle\lambda|$, we compute the II-reachable space as the least fixedpoint of the ascending chain of linear subspaces of $\mathcal{H}(\mathbb{H})$:

$$\begin{aligned}\mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| = 0\}) &\subseteq \mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| \leq 1\}) \\ &\subseteq \mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| \leq 2\}) \\ &\subseteq \cdots,\end{aligned} \tag{5}$$

where the notation $\{\mathbf{F}_\varsigma\}$ in bracket denotes a super-operator. The following lemma gives an upper bound for the occurrence of the least fixedpoint in the ascending chain.

Lemma 4.9. *Let $\Theta_0 \subseteq \Theta_1 \subseteq \Theta_2 \subseteq \cdots$ be the ascending chain of nonnull linear subspaces $\Theta_i \subseteq \mathcal{H}(\mathbb{H})$, as defined in* (5). *Then there is an integer $\ell \leq \dim(\mathbb{H})^2 - 1$ such that $\Theta_k = \Theta_\ell$ holds for all $k > \ell$.*

Proof. The proof is similar to that of Lemma 4.5. The function G from Θ_i to Θ_{i+1} ($i \geq 0$) can be formulated as a monotonic function

$$G(\mathbb{Y}) = \mathrm{span}(\mathbb{Y} \cup \{\{\mathbf{F}_\alpha\}(\gamma) : \gamma \in \mathbb{Y} \wedge \alpha \in \hat{\Sigma}\}).$$

Meanwhile, all subspaces Θ^{all} of $\mathcal{H}(\mathbb{H})$ form a complete lattice $(\Theta^{\mathrm{all}}, \subseteq, \inf, \sup)$ by taking 'inf' as the meet $\bigwedge = \bigcap$ and 'sup' as the join $\bigvee$. By Knaster–Tarski fixedpoint theorem [14, 54], we have that the least fixedpoint occurs upon $\Theta_\ell = \Theta_{\ell+1}$, where ℓ is bounded by $\dim(\mathbb{H})^2 - 1$ since Θ_i are nonnull subspaces of $\mathcal{H}(\mathbb{H})$. □

The procedure of computing the II-reachable space $\Upsilon(\mathcal{P}, \rho_0)$ is stated in Algorithm 2 with complexity analysis below.

Algorithm 2 Computing the II-Reachable Space

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$, $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ and $\mathcal{E}_j = \{\mathbf{E}_{j,k} : k = 1, 2, \ldots, K_j\}$ over $\mathbb{H}$ with dimension d, and an input pure state $\rho_0 = |\lambda\rangle\langle\lambda| \in \mathcal{D}$;
Output: a linearly independent basis θ of $\Upsilon(\mathcal{P}, \rho_0)$ whose elements are pure states.

```
let $\hat{\Sigma} = \{\alpha_{j,k} : j = 1, 2, \ldots, m \wedge k = 1, 2, \ldots, K_j\}$, and $\mathbf{E}(\alpha_{j,k}) = \mathbf{E}_{j,k}$;
let $\hat{\mathcal{P}} = (\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ be the operator-level program of $\mathcal{P}$;
$\mathbf{F}_{\alpha_{j,k}} \leftarrow \mathbf{E}_{j,k}\mathbf{M}_{\mathbf{true}}$ with $j = 1, 2, \ldots, m$ and $k = 1, 2, \ldots, K_j$;
$B_0 \leftarrow \{|\lambda\rangle\}$, $B_{-1} \leftarrow \emptyset$, and $\theta_0 \leftarrow \{\rho_0\}$;
for $i \leftarrow 1$ to $d^2 - 1$ do
    $B_i \leftarrow B_{i-1}$ and $\theta_i \leftarrow \theta_{i-1}$;
    for all $|\psi\rangle \in B_{i-1} \setminus B_{i-2}$ do
        $V \leftarrow \{\mathbf{F}_{\alpha_{j,k}} |\psi\rangle / \|\mathbf{F}_{\alpha_{j,k}} |\psi\rangle\| : j = 1, 2, \ldots, m \wedge k = 1, 2, \ldots, K_j\}$;
        find a maximal subset $B'$ of $V$, such that $\theta' = \{|\psi'\rangle\langle\psi'| : |\psi'\rangle \in B'\}$ is a linearly independent basis complement to $\theta_i$;
        $B_i \leftarrow B_i \cup B'$ and $\theta_i \leftarrow \theta_i \cup \theta'$;
    if $B_i = B_{i-1}$ or $|B_i| = d^2$ then break;
return $\theta_i$.
```

Complexity. Note that there are less than d^2 times of entering the inner loop in Line 7. Each inner loop performs at most $m \cdot d^2$ times of matrix-vector multiplication together with normalization and at most $m \cdot d^2$ times of checking the linear independence, as the factor m comes from the number of actions in $\mathcal{P}$ and the factor d^2 comes from the number of Kraus operators of $\mathcal{E}_j$. The matrix-vector multiplication is in $O(d^2)$, the normalization is in $O(d)$, and checking the linear independence can be in $O(d^4)$ with embedding into the orthonormalization of the linearly independent basis. That is, θ_i is a linearly independent basis if and only if there is an orthonormal basis ϑ_i such that $\mathrm{span}(\theta_i) = \mathrm{span}(\vartheta_i)$, in which each element can be obtained in $O(d^4)$ by the Gram–Schmit procedure. Hence Algorithm 2 is in time $O(m \cdot d^8)$. □

Example 4.10. Reconsider the program $\mathcal{P}_2$ in Example 3.7, the operator-level program $\hat{\mathcal{P}} = (\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ of $\mathcal{P}_2$ provides

- the set of actions $\hat{\Sigma} = \{\alpha_{1,1}, \alpha_{2,1}\}$;
- linear operators $\mathbf{E}(\alpha_{1,1}) = \mathbf{E}_{1,1} = \mathbf{H} \otimes \mathbf{X}$ and $\mathbf{E}(\alpha_{2,1}) = \mathbf{E}_{2,1} = \mathbf{X} \otimes \mathbf{H}$.

We define $\mathbf{F}_{\alpha_{1,1}} = \mathbf{E}_{1,1}\mathbf{M}_{\mathbf{true}}$ and $\mathbf{F}_{\alpha_{2,1}} = \mathbf{E}_{2,1}\mathbf{M}_{\mathbf{true}}$. By Algorithm 2, for the input pure state $\rho = |1,1\rangle\langle 1,1|$, the II-reachable space can be computed as follows.

(1) Initially, we have $B_0 = \{|1,1\rangle\}$ and $\theta_0 = \{|1,1\rangle\langle 1,1|\}$.
(2) Then, we compute

$$\mathbf{F}_{\alpha_{1,1}} |1,1\rangle / \|\mathbf{F}_{\alpha_{1,1}} |1,1\rangle\| = |-,0\rangle\,,$$
$$\mathbf{F}_{\alpha_{2,1}} |1,1\rangle / \|\mathbf{F}_{\alpha_{2,1}} |1,1\rangle\| = |0,-\rangle\,.$$

So we have $V = \{|-,0\rangle, |0,-\rangle\}$. Since the two pure states in V have density operators that form a linearly independent basis complement to θ_0, we obtain $B_1 = B_0 \cup V = \{|1,1\rangle, |-,0\rangle, |0,-\rangle\}$ and $\theta_1 = \{|\psi\rangle\langle\psi| : \psi \in B_1\} = \{|1,1\rangle\langle 1,1|, |-,0\rangle\langle -,0|, |0,-\rangle\langle 0,-|\}$.

(3) Repeating this process, we have

$$\begin{aligned}B_2 &= \{|1,1\rangle, |-,0\rangle, |0,-\rangle, |-,+\rangle, |+,1\rangle, |1,+\rangle\},\\ B_3 &= B_2 \cup \{(|-,0\rangle - \sqrt{2}\,|1,1\rangle)/\sqrt{3}, (\sqrt{2}\,|0,0\rangle - |1,+\rangle)/\sqrt{3}\},\\ B_4 &= B_3.\end{aligned}$$

In detail, we name the eight elements in B_3 by $|\psi_1\rangle = |1,1\rangle$, $|\psi_2\rangle = |-,0\rangle$, $|\psi_3\rangle = |0,-\rangle$, $|\psi_4\rangle = |-,+\rangle$, $|\psi_5\rangle = |+,1\rangle$, $|\psi_6\rangle = |1,+\rangle$, $|\psi_7\rangle = (|-,0\rangle - \sqrt{2}\,|1,1\rangle)/\sqrt{3}$ and $|\psi_8\rangle = (\sqrt{2}\,|0,0\rangle - |1,+\rangle)/\sqrt{3}$, whose outer product form $|\psi_i\rangle\langle\psi_i|$ are reachable states of the operator-level program $\hat{\mathcal{P}}$. The eight outer products $|\psi_i\rangle\langle\psi_i|$ make up the set θ_3, which is sufficient to linearly express any reachable pure state of $\hat{\mathcal{P}}$. For instance, $\mathbf{F}_{\alpha_{2,1}} |1,+\rangle = |0,0\rangle$ and its outer product form $|0,0\rangle\langle 0,0|$ is a reachable state of $\hat{\mathcal{P}}$, which can be linearly expressed as

$$|0,0\rangle\langle 0,0| = |\psi_1\rangle\langle\psi_1| - |\psi_2\rangle\langle\psi_2| + |\psi_6\rangle\langle\psi_6| - 3\,|\psi_7\rangle\langle\psi_7| + 3\,|\psi_8\rangle\langle\psi_8|\,.$$

So we do not necessarily put $|0,0\rangle$ into B_3, nor necessarily put $|0,0\rangle\langle 0,0|$ into θ_3, since $|0,0\rangle\langle 0,0|$ is in $\mathrm{span}(\theta_3)$. Overall, the closure of B_3 under the operators $\mathbf{F}_{\alpha_{1,1}}$ and $\mathbf{F}_{\alpha_{2,1}}$ is shown as in Table 3, implying that all linear combinations of the eight outer products $|\psi_i\rangle\langle\psi_i|$ under the operators $\mathbf{F}_{\alpha_{1,1}}$ and $\mathbf{F}_{\alpha_{2,1}}$ are also in $\mathrm{span}(\theta_3)$.

Thus the least fixedpoint of the ascending chain occurs, which yields the II-reachable space $\Upsilon(\mathcal{P}_2, \rho_0) = \mathrm{span}(\{|\psi\rangle\langle\psi| : |\psi\rangle \in B_4\})$.

It is not hard to see that $\Phi(\mathcal{P}_2, \rho_0)$ contains all pure states in $\mathbb{H}_{\mathbf{VAR}}$ while $\Upsilon(\mathcal{P}_2, \rho_0)$ has dimension 8 that is less than $\dim(\mathcal{H}(\mathbb{H}_{\mathbf{VAR}})) = 16$. Hence there are many pure states in $\Phi(\mathcal{P}_2, \rho_0)$ whose density operators are not in $\Upsilon(\mathcal{P}_2, \rho_0)$, e. g., the pure state $|\varphi\rangle\langle\varphi|$ with $|\varphi\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$ in $\Phi(\mathcal{P}_2, \rho_0)$ cannot be linearly

Table 3. The closure of B_3 under the operators $\mathbf{F}_{\alpha_{1,1}}$ and $\mathbf{F}_{\alpha_{2,1}}$

$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_1\rangle = \lvert -,0\rangle = \lvert\psi_2\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_1\rangle = \lvert 0,-\rangle = \lvert\psi_3\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_2\rangle = \lvert 1,1\rangle = \lvert\psi_1\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_2\rangle = -\lvert -,+\rangle = -\lvert\psi_4\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_3\rangle /\Vert\mathbf{F}_{\alpha_{1,1}} \lvert\psi_3\rangle\Vert = \lvert +,1\rangle = \lvert\psi_5\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_3\rangle /\Vert\mathbf{F}_{\alpha_{2,1}} \lvert\psi_3\rangle\Vert = \lvert 1,+\rangle = \lvert\psi_6\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_4\rangle /\Vert\mathbf{F}_{\alpha_{1,1}\lvert\psi_4\rangle\Vert} = \frac{\sqrt{2}\lvert 1,1\rangle - \lvert -,0\rangle}{\sqrt{3}} = -\lvert\psi_7\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_4\rangle /\Vert\mathbf{F}_{\alpha_{2,1}} \lvert\psi_4\rangle\Vert = \frac{\lvert 1,+\rangle - \sqrt{2}\lvert 0,0\rangle}{\sqrt{3}} = -\lvert\psi_8\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_5\rangle /\Vert\mathbf{F}_{\alpha_{1,1}} \lvert\psi_5\rangle\Vert = \lvert -,0\rangle = \lvert\psi_2\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_5\rangle /\Vert\mathbf{F}_{\alpha_{2,1}} \lvert\psi_5\rangle\Vert = \lvert 0,-\rangle = \lvert\psi_3\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_6\rangle = \lvert -,+\rangle = \lvert\psi_4\rangle$	$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_6\rangle = \lvert 0,0\rangle$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_7\rangle /\Vert\mathbf{F}_{\alpha_{1,1}} \lvert\psi_7\rangle\Vert = \frac{\lvert 1,1\rangle - \sqrt{2}\lvert -,0\rangle}{\sqrt{3}} = \lvert\varphi_{7,1}\rangle$	$\lvert\varphi_{7,1}\rangle\langle\varphi_{7,1}\rvert = -\frac{1}{3}\lvert\psi_1\rangle\langle\psi_1\rvert + \frac{1}{3}\lvert\psi_2\rangle\langle\psi_2\rvert + \lvert\psi_7\rangle\langle\psi_7\rvert$
$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_7\rangle /\Vert\mathbf{F}_{\alpha_{2,1}} \lvert\psi_7\rangle\Vert = \frac{-\lvert -,+\rangle - \sqrt{2}\lvert 0,-\rangle}{\sqrt{3}} = \lvert\varphi_{7,2}\rangle$	$\lvert\varphi_{7,2}\rangle\langle\varphi_{7,2}\rvert = \frac{1}{3}\lvert\psi_3\rangle\langle\psi_3\rvert - \frac{1}{3}\lvert\psi_4\rangle\langle\psi_4\rvert + \lvert\psi_8\rangle\langle\psi_8\rvert$
$\mathbf{F}_{\alpha_{1,1}} \lvert\psi_8\rangle /\Vert\mathbf{F}_{\alpha_{1,1}} \lvert\psi_8\rangle\Vert = \frac{\sqrt{2}\lvert +,1\rangle - \lvert -,+\rangle}{\sqrt{3}} = \lvert\varphi_{8,1}\rangle$	$\lvert\varphi_{8,1}\rangle\langle\varphi_{8,1}\rvert = -\frac{1}{3}\lvert\psi_4\rangle\langle\psi_4\rvert + \frac{1}{3}\lvert\psi_5\rangle\langle\psi_5\rvert + \lvert\psi_7\rangle\langle\psi_7\rvert$
$\mathbf{F}_{\alpha_{2,1}} \lvert\psi_8\rangle /\Vert\mathbf{F}_{\alpha_{2,1}} \lvert\psi_8\rangle\Vert = \frac{\sqrt{2}\lvert 1,+\rangle - \lvert 0,0\rangle}{\sqrt{3}} = \lvert\varphi_{8,2}\rangle$	$\lvert\varphi_{8,2}\rangle\langle\varphi_{8,2}\rvert = -\frac{1}{3}\lvert\psi_1\rangle\langle\psi_1\rvert + \frac{1}{3}\lvert\psi_2\rangle\langle\psi_2\rvert + \lvert\psi_7\rangle\langle\psi_7\rvert$

expressed by the basis of $\Upsilon(\mathcal{P}_2, \rho_0)$. The II-reachable space $\Upsilon(\mathcal{P}_2, \rho_0)$ gives an over-approximation of $\Psi(\mathcal{P}_2, \rho_0)$ more precise than $\Phi(\mathcal{P}_2, \rho_0)$ in this example. □

Fineness in Reachable Spaces. The ascending chain $\Theta_0 \subseteq \Theta_1 \subseteq \Theta_2 \subseteq \cdots$ as defined in (5) is finer than the ascending chain $\mathbb{S}_0 \subseteq \mathbb{S}_1 \subseteq \mathbb{S}_2 \subseteq \cdots$ as defined in (3) in such a sense:

- For each linear subspace $\Theta_i \subseteq \mathcal{H}(\mathbb{H})$, there is a unique index j such that $\Theta_i \subseteq \mathcal{H}(\mathbb{S}_j)$ and $\Theta_i \nsubseteq \mathcal{H}(\mathbb{S}_{j-1})$.
- For each linear subspace $\mathbb{S}_j \subseteq \mathbb{H}$, there is one index i or more such that $\Theta_i \subseteq \mathcal{H}(\mathbb{S}_j)$ and $\Theta_i \nsubseteq \mathcal{H}(\mathbb{S}_{j-1})$.
- By the construction in Algorithm 2 that the basis elements in Θ_i are pure states, all ensembles of elements in Θ_i are elements of $\mathcal{D}(\mathbb{S}_j)$.

In a nutshell, each increment in $\mathbb{S}_j$ corresponds to one or more increments in Θ_i. □

By Algorithms 1 and 2, we obtain the result:

Theorem 4.11. *Both I-reachable space and II-reachable space are computable in polynomial time.*

5 COMPUTING THE DIVERGENT SET

In this section, we show how the set of *divergent* states can be computed from which a given nondeterministic quantum program terminates with probability zero under some scheduler, and synthesize the corresponding divergence schedulers. The procedure turns out to be in exponential time, which as far as we know is reported for the first time.

Definition 5.1. Given a nondeterministic quantum program $\mathcal{P}$ with the quantum state space $\mathbb{H}$,

- the set $D(\mathcal{P})$ of divergent states is $\{\rho \in \mathcal{D}(\mathbb{H}) : \lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(\rho)) = 1 \wedge \sigma \in \Sigma^{\omega}\}$;
- the set $PD(\mathcal{P})$ of pure divergent states is $\{|\psi\rangle \in \mathbb{H} : \lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(|\psi\rangle\langle\psi|)) = 1 \wedge \sigma \in \Sigma^{\omega}\}$.

The parameters $\mathcal{P}$ in $D(\mathcal{P})$ and $PD(\mathcal{P})$ can be omitted if they are clear from the context.

The divergence requires that under some infinite scheduler σ, all eigenstates $|\lambda\rangle$ of ρ are terminating with probability zero, i. e. $\bigwedge_{i=0}^{\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{false}}\mathcal{F}_{\sigma\uparrow i}(|\lambda\rangle\langle\lambda|)) = 0$. It is not hard to see that an element $|\psi\rangle$ in PD is a pure state $|\psi\rangle\langle\psi|$ in D, i. e. $PD = \{|\psi\rangle \in \mathbb{H} : |\psi\rangle\langle\psi| \in D(\mathcal{P})\}$. Once the divergent set D is determined, the pure divergent set PD is also determined, which is essential. We will focus on how to compute the pure divergent set PD afterwards.

For convenience, we introduce some auxiliary notions and notations:

- PD^σ denotes the set of all pure divergent states $|\psi\rangle$ under the infinite scheduler σ, i. e.

$$PD^\sigma = \left\{ |\psi\rangle \in \mathbb{H} : \lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(|\psi\rangle\langle\psi|)) = 1 \right\};$$

- PD_i^σ denotes the set of all pure states $|\psi\rangle$ that are terminating with probability zero under the i-fragment of the infinite scheduler σ, i. e.

$$PD_i^\sigma = PD^{\sigma\uparrow i} = \{|\psi\rangle \in \mathbb{H} : \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(|\psi\rangle\langle\psi|)) = 1\};$$

- PD_i denotes the set of all pure states $|\psi\rangle$ that are terminating with probability zero under the i-fragment of some infinite scheduler σ, i. e. $PD_i = \bigcup_{\sigma\in\Sigma^\omega} PD_i^\sigma = \bigcup_{\varsigma\in\Sigma^i} PD^\varsigma$.

It is not hard to see:

- for any infinite scheduler σ and any integer i, $PD_i^\sigma \supseteq PD_{i+1}^\sigma$, as the latter requires to be terminating with probability zero for one more step, i. e. $\mathrm{tr}(\mathbf{M}_{\mathbf{false}}\mathcal{F}_{\sigma\uparrow(i+1)}(|\psi\rangle\langle\psi|)) = 0$;
- for any infinite scheduler σ, $PD^\sigma = \bigcap_{i=0}^\infty PD_i^\sigma = \lim_{i\to\infty} PD_i^\sigma$;
- for any integer i, $PD_i = \bigcup_{\sigma\in\Sigma^\omega} PD_i^\sigma$ amounts to a finite union of PD_i^σ, as there are only finitely many distinct i-fragments $\varsigma \in \Sigma^i$ of all infinite schedulers $\sigma \in \Sigma^\omega$;
- $PD = \bigcap_{i=0}^\infty PD_i = \lim_{i\to\infty} PD_i$.

From the definition $PD_i = \bigcup_{\varsigma\in\Sigma^i} PD^\varsigma$, the derivation of those pure divergent sets PD_i can be described by an infinite m-branching tree, named *derivation tree* (see Fig. 1). Its nodes are all pure divergent sets PD^ς, organized by the prefix relationship on strings $\varsigma \in \Sigma^*$. Particularly,

- the root of the tree is $PD^\epsilon = PD_0$ that is the pure divergent set under the empty scheduler ϵ;
- each intermediate node PD^ς has m children $PD^{\varsigma\cdot\alpha}$ that are the sets of pure divergent states derived by one more step $\alpha \in \Sigma$.

Thus, the union of PD^ς in the ith layer is actually PD_i. By the nice property $PD^{\sigma\uparrow i} \supseteq PD^{\sigma\uparrow(i+1)}$, we have that an intermediate node PD^ς is a common superset of its m children $PD^{\varsigma\cdot\alpha}$ with α ranging over Σ. The derivation tree is called *stabilized* at the ℓth layers if $PD_k = PD_\ell$ holds for all $k > \ell$. So the pure divergent set PD could be collected at that layer.

Besides the definition of PD_i, there is an alternative approach to calculate PD_i. Before stating it, we need to introduce the notion of *pure divergent space* PDS^ς that is the closure of PD^ς under scalar multiplication, i. e. $PDS^\varsigma = \{c\,|\psi\rangle : |\psi\rangle \in PD^\varsigma \wedge c \in \mathbb{C}\}$. The notions PDS_i and PDS are defined similarly. For a finite scheduler ς, PDS^ς is a subspace contained in $\mathbb{H}$ [47, Lemma 4][1], PD^ς is the unit sphere of PDS^ς, and they can be mutually determined. Particularly, if PDS^ς is the null space $\{0\}$, there is no element $|\psi\rangle \in PD^\varsigma$, i. e., PD^ς is the empty set; vice versa. It entails

$$PD^\varsigma = \emptyset \Longleftrightarrow PDS^\varsigma = \{0\} \Longleftrightarrow \dim(PDS^\varsigma) = 0.$$

Proposition 5.2. *The pure divergent sets PD_i can be calculated inductively as*

$$PD_i = \begin{cases} \{|\psi\rangle \in \mathbb{H} : \mathbf{M}_{\mathbf{false}}\,|\psi\rangle = 0\} & \text{if } i = 0, \\ \bigcup_{\alpha\in\Sigma}\{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS_{i-1}\} & \text{if } i > 0. \end{cases}$$

[1] To see why it is the case, we note that: PDS^ς is equivalently defined as $\{c\,|\psi\rangle : \bigwedge_{i=0}^{|\varsigma|} \mathrm{tr}(\mathbf{M}_{\mathbf{false}}\mathcal{F}_{\varsigma\uparrow i}(|\psi\rangle\langle\psi|)) = 0 \wedge |\psi\rangle \in \mathbb{H} \wedge c \in \mathbb{C}\}$, where $\mathrm{tr}(\mathbf{M}_{\mathbf{false}}\mathcal{F}_{\varsigma\uparrow i}(|\psi\rangle\langle\psi|)) = 0$ holds if and only if $\mathrm{supp}(\mathbf{M}_{\mathbf{false}})$ is orthogonal to $\mathrm{supp}(\mathcal{F}_{\varsigma\uparrow i}(|\psi\rangle\langle\psi|))$. By the inclusion (1), if $\mathrm{supp}(\mathbf{M}_{\mathbf{false}})$ is orthogonal to each $\mathrm{supp}(\mathcal{F}_{\varsigma\uparrow i}(|\psi_k\rangle\langle\psi_k|))$ $(k = 1, 2, \ldots, K)$, it is also orthogonal to $\mathrm{supp}(\mathcal{F}_{\varsigma\uparrow i}(|\psi\rangle\langle\psi|))$ for any $|\psi\rangle \in \mathrm{span}(\{|\psi_k\rangle : k = 1, 2, \ldots, K\})$, i. e. $\mathrm{tr}(\mathbf{M}_{\mathbf{false}}\mathcal{F}_{\varsigma\uparrow i}(|\psi\rangle\langle\psi|)) = 0$, yielding the linearity $|\psi\rangle \in PD^\varsigma \subset PDS^\varsigma$.

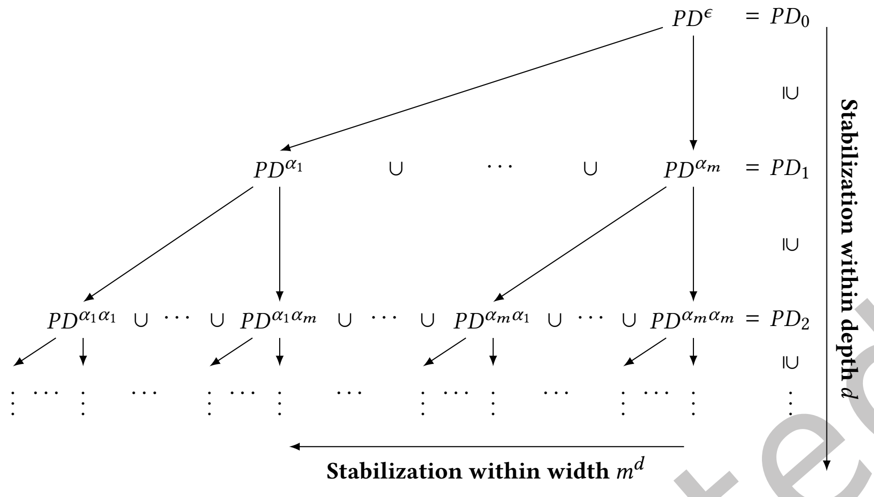
PD^{ϵ}
$= PD_0$
$\cup\!\!|$
PD^{α_1} $\cup$ $\cdots$ $\cup$ PD^{α_m}
$= PD_1$
$\cup\!\!|$
$PD^{\alpha_1\alpha_1}$ $\cup \cdots \cup$ $PD^{\alpha_1\alpha_m}$ $\cup \cdots \cup$ $PD^{\alpha_m\alpha_1}$ $\cup \cdots \cup$ $PD^{\alpha_m\alpha_m}$
$= PD_2$
$\cup\!\!|$
Stabilization within depth d
Stabilization within width m^d

Fig. 1. Derivation of PD_i by a tree construction

Proof. If $i = 0$, we have $PD_0 = PD^{\epsilon} = \{|\psi\rangle \in \mathbb{H} : \mathbf{M}_{\mathbf{false}} |\psi\rangle = 0\}$ plainly. Otherwise, for a sphere PD^{ς} in the union PD_{i-1} and an action $\alpha \in \Sigma$, we compute:

$$\begin{aligned} PD^{\alpha\cdot\varsigma} &= \{|\psi\rangle \in PD_0 : \mathcal{F}_\alpha(|\psi\rangle\langle\psi|) \in \mathcal{D}(PDS^{\varsigma})\} \\ &= \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS^{\varsigma}\}. \end{aligned} \tag{6}$$

We collect all spheres $PD^{\alpha\cdot\varsigma}$ with α ranging over Σ and ς ranging over Σ^{i-1} as PD_i, i. e.,

$$\begin{aligned} PD_i &= \bigcup_{\alpha\in\Sigma} \bigcup_{\varsigma\in\Sigma^{i-1}} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS^{\varsigma}\} \\ &= \bigcup_{\alpha\in\Sigma} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS_{i-1}\}, \end{aligned}$$

where the second equality comes from the fact that the linear subspace $\mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|))$ is covered by PDS_{i-1} if and only if it is covered by some subspace PDS^{ς} in the union PDS_{i-1}. □

As an immediate corollary, from any subtree rooted at PD^{ς} with $\varsigma \in \Sigma^*$, we can get

$$PD^{\varsigma} \cap PD_{|\varsigma|+i+1} = \bigcup_{\alpha\in\Sigma} \{|\psi\rangle \in PD^{\varsigma} : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq (PDS^{\varsigma} \cap PDS_{|\varsigma|+i})\},$$

where $PD^{\varsigma} \cap PD_{|\varsigma|+i}$ denotes the union of $PD^{\varsigma\cdot\varsigma'}$ over $\varsigma' \in \Sigma^i$.

By the alternative approach, the set PD_i is calculated from the prior set PD_{i-1}, which will be used to establish the stabilization of derivation tree. That is, an upper bound is given below for the occurrence of the least fixedpoint in the descending chain of finite unions PD_i, one-to-one corresponding to PDS_i.

Lemma 5.3. *Let $PDS_0 \supseteq PDS_1 \supseteq PDS_2 \supseteq \cdots$ be a descending chain of finite unions of nonempty subspaces $PDS_i \subseteq \mathbb{H}$, as calculated in Proposition 5.2. Then there is an integer $\ell \le d$ such that $PDS_k = PDS_\ell$ holds for all $k > \ell$.*

Proof. The proof improves that of [47, Lemma 6] by giving the explicit bound d. It is crucial in establishing the complexity of computing the pure divergent set, which is left as an open problem in [47, Subsection 7.3]. We complete the proof by an induction on the dimension d_0 of PDS_0.

- Basically, when $d_0 = 0$, we have $PDS_0 = \{0\}$. It is plainly the fixedpoint of the chain, implying the pure divergent set PD is empty then.
- Inductively, when $d_0 > 0$, we first tackle the case $PDS_0 \supset PDS_1$. Let $PDS_1 = \bigcup_{\alpha\in\Sigma} PDS^\alpha$ where PDS^α are subspaces in the union. Define $Z_{\alpha,i} = PDS^\alpha \cap PDS_{1+i}$ for $i \geq 0$. We have $PDS_{1+i} = \bigcup_{\alpha\in\Sigma} Z_{\alpha,i}$ and the following m descending chains:

  $$PDS^\alpha = Z_{\alpha,0} \supseteq Z_{\alpha,1} \supseteq Z_{\alpha,2} \supseteq \cdots \qquad \text{for } \alpha \in \Sigma.$$

  Since PDS_0 is a single subspace, it follows $\dim(PDS^\alpha) < d_0$ by $PDS_0 \supset PDS_1$. By induction hypothesis, there is an $\ell_\alpha \leq \dim(PDS^\alpha)$ in the respective descending chain, such that $Z_{\alpha,k} = Z_{\alpha,\ell_\alpha}$ holds for all $k > \ell_\alpha$. Thereby, we obtain that $PDS_\ell = PDS_{\ell+1}$ holds with $\ell = 1 + \max_{\alpha\in\Sigma} \dim(PDS^\alpha) \leq d_0 \leq d$. We then tackle the case $PDS_0 = PDS_1$, where there is an action $\alpha_0 \in \Sigma$ such that $PDS_0 = PDS^{\alpha_0}$. We have $PDS_i = PDS^{\alpha_0^i} = PDS_0$ for $i > 0$, leading to $\ell = 0 < d_0 \leq d$ directly. We further claim that the least fixedpoint of the original descending chain occurs upon $PDS_\ell = PDS_{\ell+1}$, since

  $$\begin{aligned} PD_{\ell+2} &= \bigcup_{\alpha\in\Sigma} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS_{\ell+1}\} \\ &= \bigcup_{\alpha\in\Sigma} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS_{\ell}\} \\ &= PD_{\ell+1} = PD_\ell \end{aligned}$$

  and $PD_k = PD_\ell$ follows for all $k > \ell + 2$ similarly. □

The above lemma indicates that the derivation tree can be stabilized within height d and width m^d. The procedure of computing the pure divergent set PD is stated in Algorithm 3. In detail, Sch_{i-1} in Line 4 stores those finite schedulers ς corresponding to the nodes PD^ς to be derived. We attempt to derive the node PD^ς in Line 8, provided that the condition $\bigcup_{\varsigma'\in Sch'} PD^{\varsigma'} = PD^\varsigma$ in Line 7 is not met. Otherwise, the derivation is unnecessary since the subtree rooted at PD^ς is stabilized then. The output is a union of various Sch_i consisting of schedulers that generate PD^ς stabilized at the ith layer. The complexity of Algorithm 3 is provided below.

Algorithm 3 Computing the Pure Divergent Set

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ over $\mathbb{H}$ with dimension d;
Output: a set Sch of finite schedulers that generate the pure divergent set PD of $\mathcal{P}$.

```
1:  $Sch_0 \leftarrow \{\epsilon\}$, and compute $PD^\epsilon$;
2:  for $i \leftarrow 1$ to $d-1$ do
3:      $Sch_i \leftarrow \emptyset$ and $Sch \leftarrow \emptyset$;
4:      while $Sch_{i-1} \setminus Sch \neq \emptyset$ do
5:          let $\varsigma$ be an element of $Sch_{i-1} \setminus Sch$;
6:          $Sch' \leftarrow \{\varsigma \cdot \alpha : \alpha \in \Sigma\}$, and compute $PD^{\varsigma'}$ for each $\varsigma' \in Sch'$;
7:          if $\bigcup_{\varsigma'\in Sch'} PD^{\varsigma'} = PD^\varsigma$ then $Sch \leftarrow Sch \cup \{\varsigma\}$;
8:          else $Sch_i \leftarrow Sch_i \cup Sch'$ and $Sch_{i-1} \leftarrow Sch_{i-1} \setminus \{\varsigma\}$;
9:      if $Sch_i = \emptyset$ then break;
10: return $Sch = Sch_0 \cup Sch_1 \cup \cdots \cup Sch_{i-1}$.
```

Complexity. Note that there are at most $1 + m + \cdots + m^{d-1}$ times of entering the inner loop in Line 4. Each inner loop computes m spheres $PD^{\varsigma'}$ in Line 6, which is finished in such a way:

(1) Write $\varsigma' \in \Sigma^i$ as the form $\alpha \cdot \varsigma''$ for some $\alpha \in \Sigma$ and $\varsigma'' \in \Sigma^{i-1}$.
(2) By the last loop, we have determined the linear subspace $PDS^{\varsigma''}$ as well as the orthonormal basis $\{|\psi_j^\perp\rangle : j = 1, 2, \ldots, J\}$ of its complement where $J < d - i + 1$.
(3) Let $\{\mathbf{F}_k : k = 1, 2, \ldots, K\}$ be the Kraus representation of $\mathcal{F}_\alpha$ where $K \leq d^2$.
(4) By Eq. (6), $PD^{\varsigma'}$ is obtained as the solution space of $\mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PDS^{\varsigma''}$, i. e.,

$$\bigwedge_{j=1}^{J} \bigwedge_{k=1}^{K} \langle\psi_j^\perp| \, \mathbf{F}_k \, |\psi\rangle = 0.$$

It performs $J \cdot K$ times of matrix-vector multiplication $\langle\psi_j^\perp| \, \mathbf{F}_k$, each of which is done in $O(d^2)$, and solves $J \cdot K$ linear equations, w. r. t. d complex variables introduced to encode the pure state $|\psi\rangle$, in $O(J \cdot K \cdot d^2) \subseteq O(d^5)$.

Hence Algorithm 3 is in exponential time $O(m^d \cdot d^5)$, where the growth in the derivation tree is the bottleneck. □

Scheduler Synthesis. For each finite scheduler ς in the output *Sch* of Algorithm 3, we know there is an action $\alpha \in \Sigma$ satisfying $PD^{\varsigma} = PD^{\varsigma \cdot \alpha}$. Hence the ω-regular scheduler $\sigma = \varsigma \cdot \alpha^\omega$ is a *divergence scheduler*, under which all states on PDS^{ς} are terminating with probability zero. □

Example 5.4. Here we will compute the set *Sch* of finite schedulers that generate the pure divergent set *PD* of the program $\mathcal{P}_2$ in Example 3.7. Algorithm 3 delivers the inductive process.

(1) Initially, in the 0th layer of the derivation tree, we have $Sch_0 = \{\epsilon\}$ and

$$PDS_0 = PDS^{\epsilon} = \mathrm{span}(\{|0,0\rangle, |1,0\rangle, |1,1\rangle\})$$

to be derived.

(2) In the first layer, we derive PDS^{ϵ} for actions α_1 and α_2, and get

$$PDS^{\alpha_1} = \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}),$$
$$PDS^{\alpha_2} = \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}),$$

which are both proper subspaces of PDS^{ϵ}. So we update Sch_0 to $\emptyset$, and set $Sch_1 = \{\alpha_1, \alpha_2\}$ and $PDS_1 = PDS^{\alpha_1} \cup PDS^{\alpha_2}$ to be derived.

(3) In the second layer, we derive PDS^{α_1} and PDS^{α_2} for actions α_1 and α_2, and get

$$\begin{aligned} PDS^{\alpha_1\alpha_1} &= \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}), \\ PDS^{\alpha_2\alpha_1} &= \mathrm{span}(\{\tfrac{1}{\sqrt{3}}(-\sqrt{2}\,|0,0\rangle + |1,+\rangle)\}), \\ PDS^{\alpha_1\alpha_2} &= \mathrm{span}(\{\tfrac{1}{\sqrt{3}}(\sqrt{2}\,|1,1\rangle - |-,0\rangle)\}), \\ PDS^{\alpha_2\alpha_2} &= \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}). \end{aligned}$$

Since $PDS^{\alpha_1} = PDS^{\alpha_1\alpha_1}$ and $PDS^{\alpha_2} = PDS^{\alpha_2\alpha_2}$, the derivation subtrees rooted at them are stabilized then, as well as the whole derivation tree, i. e. $PDS_2 = PDS^{\alpha_1\alpha_1} \cup PDS^{\alpha_2\alpha_1} \cup PDS^{\alpha_1\alpha_2} \cup PDS^{\alpha_2\alpha_2} = PDS_1$.

Hence, $PD = PD_1$ is the least fixedpoint of the descending chain. We report it by the set of finite schedulers $Sch = Sch_0 \cup Sch_1 = \emptyset \cup \{\alpha_1, \alpha_2\} = \{\alpha_1, \alpha_2\}$. Additionally, we have the divergence schedulers α_1^ω for those states on PDS^{α_1} and α_2^ω for those states on PDS^{α_2}. □

By Algorithm 3 and the transformation $D = \bigcup_{PD^{\varsigma} \in PD} \mathcal{D}(PDS^{\varsigma})$, we obtain the result:

Theorem 5.5. *Both pure divergent set and divergent set are computable in exponential time.*

6 DECIDING THE TERMINATION PROBLEM

Combining the (pure) divergent sets with the reachable spaces obtained in the previous sections, we are able to decide the termination of the nondeterministic quantum programs. Although the reachable spaces are supersets of the reachable set, they could still be utilized to yield a necessary and sufficient condition to the termination as the following result.

Lemma 6.1. *Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho_0 \in \mathcal{D}$, $\mathcal{P}$ terminates with probability less than one under some scheduler if and only if the I-reachable space $\Phi(\mathcal{P}, \rho_0)$ and the pure divergent set $PD(\mathcal{P})$ are not disjoint.*

Proof. We first prove the "if" direction. Let $|\psi\rangle$ be a pure divergent state in the I-reachable space $\Phi(\mathcal{P}, \rho_0)$, which is terminating with probability zero under some scheduler σ, i. e.,

$$\lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(|\psi\rangle\langle\psi|)) = 1.$$

Let $\sigma = \varsigma \cdot \alpha^{\omega}$ be an ω-regular scheduler as the output of Algorithm 3. Since $\Phi(\mathcal{P}, \rho_0) = \bigvee_{\gamma\in\Psi(\mathcal{P},\rho_0)} \mathrm{supp}(\gamma)$ where $\Psi(\mathcal{P}, \rho_0)$ is the reachable set, there is a finite set of pure reachable states $|\psi_k\rangle$ $(k = 1, 2, \ldots, K)$ respectively in the supports of reachable states $\gamma_k \in \Psi(\mathcal{P}, \rho_0)$ reached from ρ_0 under finite schedulers ς_k, such that $|\psi\rangle = \sum_{k=1}^{K} c_k |\psi_k\rangle$ holds for some $c_k \in \mathbb{C}$. We claim that at least one, e. g., $|\psi_{k_0}\rangle$, among these $|\psi_k\rangle$ $(k = 1, 2, \ldots, K)$ is terminating with probability less than one under the scheduler σ. (Otherwise all $|\psi_k\rangle$ are terminating with probability one under σ, as well as the mixture $\gamma = \frac{1}{K}\sum_{k=1}^{K} |\psi_k\rangle\langle\psi_k|$. Since $|\psi\rangle$ is in the support $\mathrm{span}(\{|\psi_k\rangle : k = 1, 2, \ldots, K\})$ of γ, by [56, Exercise 2.73], there is a minimal probabilistic ensemble of γ containing $|\psi\rangle$ with positive probability. Then we reaches the contradiction that $|\psi\rangle$ is terminating with probability one under σ.) Therefore $\mathcal{P}$ terminates with probability less than one under the nontermination scheduler $\varsigma_{k_0} \cdot \sigma$. The workflow is shown in Fig. 2.

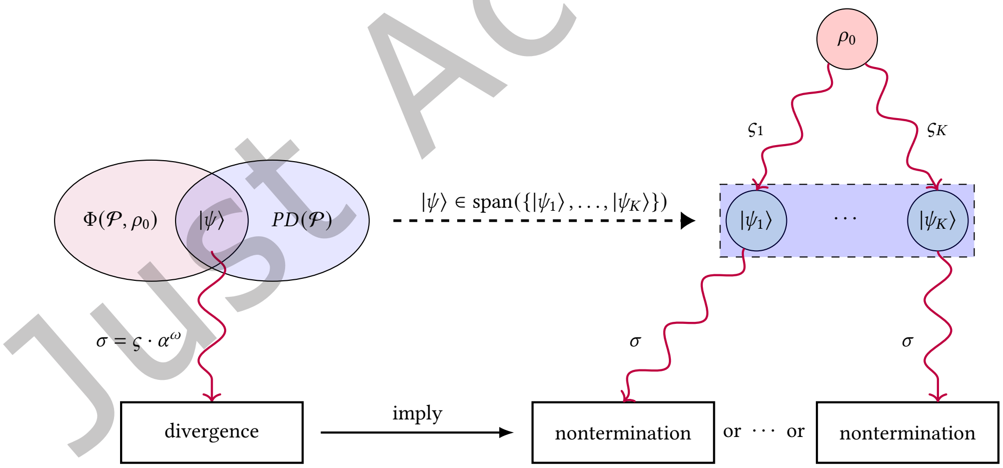

Fig. 2. The workflow of "if" direction

For the "only if" direction, we assume that σ is the nontermination scheduler, under which $\mathcal{P}$ does not terminate with probability one on ρ_0. From the input state ρ_0, $\mathcal{P}$ terminates with probability less than one. Then we will construct a sequence of pure reachable states as:

- fixed a spectral decomposition of ρ_0, there is an eigenstate $|\lambda_0\rangle$ among eigenstates in the decomposition that maximizes the nontermination probability
$$p_0 = \lim_{i\to\infty} \mathrm{tr}(\mathbf{M_{true}} \cdot \mathcal{F}_{\sigma\uparrow i}(|\lambda_0\rangle\langle\lambda_0|));$$
- fixed a spectral decomposition of $\mathcal{F}_{\sigma\uparrow 1}(|\lambda_0\rangle\langle\lambda_0|)$, there is an eigenstate $|\lambda_1\rangle$ that maximizes the nontermination probability
$$p_1 = \lim_{i\to\infty} \mathrm{tr}(\mathbf{M_{true}} \mathcal{F}_{(\sigma\downarrow 1)\uparrow i}(|\lambda_1\rangle\langle\lambda_1|));$$
- fixed a spectral decomposition of $\mathcal{F}_{\sigma\uparrow 1}(|\lambda_1\rangle\langle\lambda_1|)$, there is an eigenstate $|\lambda_2\rangle$ that maximizes the nontermination probability
$$p_2 = \lim_{i\to\infty} \mathrm{tr}(\mathbf{M_{true}} \mathcal{F}_{(\sigma\downarrow 2)\uparrow i}(|\lambda_2\rangle\langle\lambda_2|));$$
- and so on.

The nontermination probabilities $p_0, p_1, p_2, \ldots$ are monotonously increasing and convergent to some limit value p^*. We proceed to show $p^* = 1$. For any $|\lambda_i\rangle$ with nontermination probability $p_i < 1$, we know the termination probability of $|\lambda_i\rangle$ under the infinite scheduler $\sigma \downarrow i$ is $1 - p_i$, and there is a finite fragment $(\sigma \downarrow i) \uparrow j$ of $\sigma \downarrow i$ under which the termination probability of $|\lambda_i\rangle$ is at least $\frac{1}{2}(1 - p_i)$, i. e. $\mathrm{TP}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|) \geq \frac{1}{2}(1 - p_i)$. By choosing the eigenstate $|\lambda_{i+j}\rangle$ in that sequence, we know that the nontermination probability p_{i+j} of $|\lambda_{i+j}\rangle$ is not less than the average nontermination probability of $\mathcal{F}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|)$, i. e. the nontermination probability of the normalized $\mathcal{F}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|)/\mathrm{tr}(\mathcal{F}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|))$. The nontermination probability of $\mathcal{F}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|)$ is still p_i, while the trace $\mathrm{tr}(\mathcal{F}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|))$ is $1 - \mathrm{TP}_{(\sigma\downarrow i)\uparrow j}(|\lambda_i\rangle\langle\lambda_i|) \leq \frac{1}{2}(1 + p_i)$. The average nontermination probability is at least $2p_i/(1 + p_i)$, which is also a lower bound of the nontermination probability p_{i+j} of $|\lambda_{i+j}\rangle$. So we have $p_{i+1} \geq 2p_i/(1 + p_i)$. Taking the limit, we get $p^* \geq 2p^*/(1 + p^*)$, which entails $p^* = 1$.

Those eigenstates $|\lambda_0\rangle, |\lambda_1\rangle, |\lambda_2\rangle, \ldots$ are unit vectors in the I-reachable subspace $\Phi(\mathcal{P}, \rho)$ of $\mathbb{H}$. We have that there is a convergent subsequence $|\lambda'_1\rangle, |\lambda'_2\rangle, |\lambda'_3\rangle, \ldots$ also in the I-reachable subspace $\Phi(\mathcal{P}, \rho)$. By the completeness of finite-dimensional Hilbert space that the limit of a convergent sequence is contained in that space, the limit $|\lambda'\rangle$ of the subsequence $|\lambda'_1\rangle, |\lambda'_2\rangle, |\lambda'_3\rangle, \ldots$ is in the finite-dimensional Hilbert space $\Phi(\mathcal{P}, \rho)$, which is a pure divergent state as $|\lambda'\rangle$ has the nontermination probability $p^* = 1$. □

The above proof only tells us that at least one pure reachable state $|\psi_{k_0}\rangle$ among finitely many ones $|\psi_k\rangle$ $(k = 1, 2, \ldots, K)$ is terminating with probability less than one, but does not identify it. In the following, we will identify this $|\psi_{k_0}\rangle$ by exhaustively checking whether $|\psi_k\rangle$ is terminating with probability less than one. Conditioning on the nontermination under the ω-regular scheduler $\sigma = \varsigma \cdot \alpha^\omega$, we get the following equivalent statements:

(1) $|\psi_k\rangle$ is terminating with probability less than one.
(2) $\rho = \mathcal{F}_\varsigma(|\psi_k\rangle\langle\psi_k|)$ is terminating with probability less than one.
(3) Let $\mathcal{P}^{\alpha^\omega}$ be the program $\mathcal{P}$ under the scheduler α^ω, and $\mathbb{S}$ the I-reachable space $\Phi(\mathcal{P}^{\alpha^\omega}, \rho)$. Then there is a Hermitian operator γ on $\mathbb{S}$ such that $\mathcal{F}_\alpha(\gamma) = \gamma$.

The first two statements are equivalent since $|\psi_k\rangle\langle\psi_k|$ and ρ have the same nontermination probability. The necessity of the last statement follows from Brouwer's fixedpoint theorem [33, Chapter 4], since $\mathcal{F}_\alpha$ is a continuous function from the divergent set on $\mathbb{S}$ to itself, where the divergent set on $\mathbb{S}$ is convex and compact in the viewpoint of probabilistic ensemble. The sufficiency follows from the fact that for any pure state $|\psi'\rangle$ in the support of γ satisfying $\mathcal{F}_\alpha(\gamma) = \gamma$, $\mathrm{supp}(\mathcal{F}_\alpha(|\psi'\rangle\langle\psi'|))$ is contained in that $\mathrm{supp}(\gamma)$, implying $|\psi'\rangle$ is a pure divergent state; $|\psi'\rangle$ can be linearly expressed by finitely many pure states $|\psi'_k\rangle$ $(k = 1, 2, \ldots, K')$ reachable from ρ, at least one among which is terminating with probability less than one. The workflow is shown in Fig. 3.

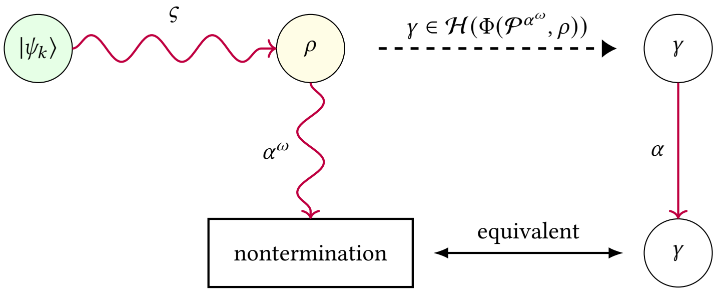

Fig. 3. The workflow of checking the nontermination of $|\psi_k\rangle$

We summarize the procedure of synthesizing a nontermination scheduler as Algorithm 4, whose complexity analysis is provided below.

Algorithm 4 Synthesizing a Scheduler for Nontermination

```
Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ over $\mathbb{H}$ with dimension $d$, and an input
    pure state $\rho_0 \in \mathcal{D}(\mathbb{H})$;
Output: a scheduler under which $\mathcal{P}$ terminates with probability less than one on $\rho_0$ if exists.
 1: compute the I-reachable space $\Phi(\mathcal{P}, \rho_0)$ by Algorithm 1;
 2: compute the pure divergent set $PD(\mathcal{P})$ by Algorithm 3;
 3: if $\Phi(\mathcal{P}, \rho_0) \cap PD(\mathcal{P}) \neq \emptyset$ then
 4:     let $|\psi\rangle$ be an element in $\Phi(\mathcal{P}, \rho_0) \cap PD^{\varsigma}$ for some $PD^{\varsigma} \Subset PD(\mathcal{P})$;
 5:     let $\sigma = \varsigma \cdot \alpha^\omega$ be a divergence scheduler of $|\psi\rangle$;
 6: else return $\epsilon$;                                   ▷ report no nontermination scheduler
 7: let $\{|\psi_k\rangle : k = 1, 2, \ldots, K\}$ be a minimal set of pure reachable states under schedulers $\varsigma_k$ that linearly
    express $|\psi\rangle$;
 8: for $k \leftarrow 1$ to $K$ do                    ▷ exhaustively checking $|\psi_k\rangle$ for nontermination
 9:     $\rho \leftarrow \mathcal{F}_{\varsigma}(|\psi_k\rangle\langle\psi_k|)$;
10:     compute the I-reachable space $\mathbb{S} = \Phi(\mathcal{P}^{\alpha^\omega}, \rho)$ by Algorithm 1;
11:     if $\mathcal{F}_\alpha(\gamma) = \gamma$ has some nonzero solution $\gamma \in \mathcal{H}(\mathbb{S})$ then return $\varsigma_k \cdot \sigma$.
```

Complexity. Computing $\Phi(\mathcal{P}, \rho_0)$ is in $O(m \cdot d^5)$, and computing $PD(\mathcal{P})$ is in $O(m^d \cdot d^5)$. The emptiness of $\Phi(\mathcal{P}, \rho) \cap PD(\mathcal{P})$ in Line 3 can be checked by computing whether the intersection of $\Phi(\mathcal{P}, \rho)$ and PDS^{ς} is null for each individual sphere PD^{ς} in the union $PD(\mathcal{P})$, which is in at most $m^d \times O(d^3)$. Once an element $|\psi\rangle$ in $\Phi(\mathcal{P}, \rho) \cap PD(\mathcal{P})$ is obtained, we can find finitely many pure states $|\psi_k\rangle$ ($k = 1, 2, \ldots, K$) to linearly express $|\psi\rangle$, which has been embedded into the computation of $\Phi(\mathcal{P}, \rho_0)$. There are at most $K \leq d$ times of entering the loop in Line 8. Each loop

(1) performs $\mathcal{F}_{\varsigma}(|\psi_k\rangle\langle\psi_k|)$ in $O(d^5)$, since it is $|\varsigma| \leq d$ times of performing quantum operations on density operators;
(2) computes $\Phi(\mathcal{P}^{\alpha^\omega}, \rho)$ which is in $O(d^5)$ since the action set of $\mathcal{P}^{\alpha^\omega}$ is a singleton set $\{\alpha\}$;
(3) solves $\mathcal{F}_\alpha(\gamma) = \gamma$ which is in $O(d^6)$ since it is a system of linear equations in d^2 real variables for encoding the Hermitian operator $\gamma \in \mathcal{H}(\mathbb{S})$.

Hence Algorithm 4 is in exponential time $O(m^d \cdot d^5 + d^7)$, whose bottleneck lies in the computation of $PD(\mathcal{P})$. □

Example 6.2. For the while-loop $\mathcal{P}_2$ in Example 3.7 with input state $\rho_0 = |1,1\rangle\langle 1,1|$, we have obtained the I-reachable subspace $\Phi(\mathcal{P}_2, \rho_0) = \mathbb{H}_{\mathbf{VAR}}$ and the pure divergent set $PD(\mathcal{P}_2) = PD^{\alpha_1} \cup PD^{\alpha_2}$ with $PDS^{\alpha_1} = \mathrm{span}(\{|1,1\rangle, |-,0\rangle\})$ and $PDS^{\alpha_2} = \mathrm{span}(\{|0,0\rangle, |1,+\rangle\})$, on which the divergence schedulers are $\sigma_i = \alpha_i^\omega$ respectively, in the previous examples.

The intersection of $\Phi(\mathcal{P}_2, \rho_0)$ and $PD(\mathcal{P}_2)$ is not empty, as it has elements $|1,1\rangle$, $|-,0\rangle$, $|0,0\rangle$ and $|1,+\rangle$. It is clear that $|1,1\rangle$ and $|-,0\rangle$ are pure reachable states respectively in the supports of ρ_0 and $\mathcal{F}_{\alpha_1}(\rho_0) = \frac{1}{2}|-,0\rangle\langle -,0|$. To demonstrate the generality of our method, we exemplify it with $|0,0\rangle$ to find out a pure reachable state that is terminating with probability less than one. Since $|0,0\rangle = |1,1\rangle - \sqrt{2}\,|-,0\rangle + \sqrt{2}\,|0,-\rangle + 2\,|-,+\rangle$ is linearly expressed by the pure reachable states $|1,1\rangle$ under the finite scheduler $\varsigma_1 = \epsilon$, $|-,0\rangle$ under $\varsigma_2 = \alpha_1$, $|0,-\rangle$ under $\varsigma_3 = \alpha_2$ and $|-,+\rangle$ under $\varsigma_4 = \alpha_1\alpha_2$, we know that at least one among $|1,1\rangle$, $|-,0\rangle$, $|0,-\rangle$ and $|-,+\rangle$ is terminating with probability less than one.

The two pure states $|1,1\rangle$ and $|-,0\rangle$ are divergent, thus they are terminating with probability zero. Again, to demonstrate our method, we will check whether $|-,+\rangle$ is terminating with probability less than one as follows. The I-reachable subspace $\Phi(\mathcal{P}_2^{\sigma_2}, |-,+\rangle\langle -,+|)$ is $\mathbb{S} = \mathrm{span}(\{|-,+\rangle, |0,-\rangle, |1,+\rangle\})$. Solving $\mathcal{F}_{\alpha_2}(\gamma) = \gamma$ with $\gamma \in \mathcal{H}(\mathbb{S})$, we get a nonzero solution $|\phi\rangle\langle\phi|$ with $|\phi\rangle = |-,+\rangle + |0,-\rangle/\sqrt{2} + (1+\sqrt{2})\,|1,+\rangle/\sqrt{2}$. Hence the nontermination scheduler $\varsigma_4 \cdot \sigma_2$ is synthesized to force $\mathcal{P}_2$ to terminate with probability less than one on ρ_0, which entails the protocol is defective. □

By a similar analysis on the II-reachable space and the divergent set, we get:

Corollary 6.3. *Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho_0 = |\lambda_0\rangle\langle\lambda_0| \in \mathcal{D}$, $\mathcal{P}$ terminates with probability less than one on ρ_0 under some scheduler if and only if the II-reachable space $\Upsilon(\mathcal{P}, \rho_0)$ and the divergent set $D(\mathcal{P})$ are not disjoint.*

Proof. We only prove the "if" direction, while the "only if" direction is the same as that of Lemma 6.1. Let ρ be a divergent state in the II-reachable space $\Upsilon(\mathcal{P}, \rho_0)$, which has no probability of termination under some scheduler, e. g., σ, i. e., $\lim_{i\to\infty} \mathrm{tr}(\overline{\mathbf{M}}_{\mathbf{nt}}\mathcal{F}_{\sigma\uparrow i}(\rho)) = 1$. Since $\Upsilon(\mathcal{P}, \rho_0) = \bigvee_{\gamma\in\Psi(\hat{\mathcal{P}},\rho_0)} \mathrm{supp}(\gamma)$ where $\hat{\mathcal{P}}$ is the operator-level program of $\mathcal{P}$ and $\Psi(\hat{\mathcal{P}}, \rho_0)$ is the reachable set of $\hat{\mathcal{P}}$, there is a finite set of pure reachable states $|\psi_k\rangle\langle\psi_k|$ ($k = 1, 2, \ldots, K$), respectively reached from ρ_0 under the finite scheduler ς_k, such that $\rho = \sum_{k=1}^K c_k\,|\psi_k\rangle\langle\psi_k|$ holds for some $c_k \in \mathbb{R}$. We claim that at least one, e. g., $|\psi_{k_0}\rangle\langle\psi_{k_0}|$, among these $|\psi_k\rangle\langle\psi_k|$ ($k = 1, 2, \ldots, K$) has a positive probability of nontermination under the scheduler σ, since otherwise all $|\psi_k\rangle\langle\psi_k|$ have no probability of nontermination under σ, as well as their linear combination ρ. Therefore $\mathcal{P}$ does not terminate with probability one under the scheduler $\varsigma_{k_0} \cdot \sigma$. □

Algorithm 4 could be amended to Lemma 6.3 by checking the emptiness of $\Upsilon(\mathcal{P}, \rho_0) \cap D(\mathcal{P})$. To this end, for each individual sphere PD^ς in $PD(\mathcal{P})$, we have to

(1) introduce $\dim(PDS^\varsigma) \le d$ complex variables to encode a pure state $|\psi\rangle \in PD^\varsigma$,
(2) introduce $\dim(\Upsilon(\mathcal{P}, \rho_0)) \le d^2$ real variables to encode an element in $\Upsilon(\mathcal{P}, \rho_0)$,
(3) $|\psi\rangle\langle\psi| \in \Upsilon(\mathcal{P}, \rho_0)$ results in a polynomial formula in those variables, whose coefficients are algebraic numbers. It can be solved in $2^{O(d^2)}$ by the existential theory of the reals [7, Theorem 13.13].

Hence it would contribute an additional factor $2^{O(d^2)}$ to the complexity of the procedure.

Example 6.4. For the while-loop $\mathcal{P}_2$ in Example 3.7 with input state $\rho_0 = |1,1\rangle\langle 1,1|$, we have obtained the II-reachable subspace

$$\Upsilon(\mathcal{P}_2, \rho_0) = \mathrm{span}\left(\left\{|\psi\rangle\langle\psi| : |\psi\rangle \in \left\{\begin{array}{l} |1,1\rangle, |-,0\rangle, |0,-\rangle, |-,+\rangle, |+,1\rangle, |1,+\rangle, \\ (\sqrt{2}\,|0,0\rangle - |1,+\rangle)/\sqrt{3}, (|-,0\rangle - \sqrt{2}\,|1,1\rangle)/\sqrt{3} \end{array}\right\}\right\}\right)$$

and the divergent set $D(\mathcal{P}_2) = \mathcal{D}(PDS^{\alpha_1}) \cup \mathcal{D}(PDS^{\alpha_2})$ with $PDS^{\alpha_1} = \operatorname{span}(\{|1,1\rangle, |-,0\rangle\})$ and $PDS^{\alpha_2} = \operatorname{span}(\{|0,0\rangle, |1,+\rangle\})$ in the previous examples. They have common elements such as $|1,1\rangle\langle 1,1|$, which also refutes the termination. □

Theorem 6.5. *The termination problem described in Problem 3.11 can be solved in exponential time.*

7 SYNTHESIZING A UNIVERSAL SCHEDULER

In this section, we study the universal termination problem, which asks whether all input states of a program are terminating with probability one under their respective schedulers. We first decide the universal termination by detecting the existence of invariant subspace contained in $\mathbb{H}$. If the answer is affirmative, we could further synthesize a universal scheduler, which forces all input states to be terminating with probability one. The procedure turns out to be in polynomial time, which is also reported for the first time.

For a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$, the states to be analyzed are those density operators on the subspace $\mathbf{M_{true}}(\mathbb{H})$, the null space of $\mathbf{M_{false}}$. Thus we propose:

Definition 7.1 (Invariant Space). Given a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M_{true}}, \mathbf{M_{false}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \dots, m\}$ and $\mathcal{E}(\alpha_j) = \mathcal{E}_j$, an invariant space of $\mathcal{P}$ is a nonnull subspace $\mathbb{I}$ of $\mathbf{M_{true}}(\mathbb{H})$, satisfying that $\mathcal{E}_j(\rho) \in \mathcal{D}(\mathbb{I})$ holds for all input states $\rho \in \mathcal{D}(\mathbb{I})$ and all actions $\alpha_j \in \Sigma$.

From the above definition, we can see that the invariant subspaces $\mathbb{I}$ of $\mathbf{M_{true}}(\mathbb{H})$ have the joint semi-lattice structure with ascending chain condition. That is, for two invariant subspaces $\mathbb{I}_1$ and $\mathbb{I}_2$ of $\mathbb{H}$, the join $\mathbb{I}_1 \bigvee \mathbb{I}_2$ is also an invariant subspace; there is no infinite times of increment in the ascending chain due to the finite dimension in $\mathbb{H}$. Additionally, the invariant space $\mathbb{I}$ requires $\mathcal{E}_j(\mathcal{D}(\mathbb{I})) \subseteq \mathcal{D}(\mathbb{I})$ holds for all $\alpha_j \in \Sigma$, entailing $\bigvee_{\rho \in \mathcal{D}(\mathbb{I})} \bigvee_{j=1}^{m} \operatorname{supp}(\mathcal{E}_j(\rho)) \subseteq \mathbb{I}$. Define a function E on linear subspaces $\mathbb{S}$ of $\mathbb{H}$ as:

$$E(\mathbb{S}) := \bigvee_{\rho \in \mathcal{D}(\mathbb{S})} \bigvee_{j=1}^{m} \operatorname{supp}(\mathcal{E}_j(\rho)). \tag{7}$$

It is a monotonic function. For any invariant space $\mathbb{I}$, thank to Knaster–Tarski fixedpoint theorem [14, 54], we know there is a greatest fixedpoint $\mathbb{I}_0 \subseteq \mathbb{I}$ such that $E(\mathbb{I}_0) = \mathbb{I}_0$. So we would refer the invariant space $\mathbb{I}$ as the greatest fixedpoint $\mathbb{I}_0$ of the function E afterwards.

The existence of invariant space $\mathbb{I}$ implies that $\mathcal{P}$ terminates on those states $\rho \in \mathcal{D}(\mathbb{I})$ with probability zero, no matter which scheduler is taken. What is more important is the converse:

Lemma 7.2. *If there is an input state on which the program $\mathcal{P}$ terminates with probability less than one under any scheduler, $\mathcal{P}$ has an invariant space $\mathbb{I}$.*

Proof. Let $\mathbb{S}_0$ be the subspace $\mathbf{M_{false}}(\mathbb{H})$. We prove this lemma by contradiction. Assume that $\mathcal{P}$ has no invariant subspace, meaning no invariant subspace is contained in $\mathbf{M_{true}}(\mathbb{H}) = \mathbb{S}_0^{\perp}$. By [71, Theorem 7], we have that there is no invariant subspace contained in $\mathbb{S}_0^{\perp}$ if and only if there is a scheduler such that the probability of reaching $\mathbb{S}_0$ is 1 for all initial states $\rho \in \mathcal{D}(\mathbb{S}_0^{\perp})$. It entails that there is a scheduler under which $\mathcal{P}$ terminates with probability 1 for all input states, which contradicts the premise. Hence, under the premise $\mathcal{P}$ has an invariant subspace. □

To efficiently compute the invariant space $\mathbb{I}$ defined as the greatest fixedpoint of (7), we will derive a series of necessary conditions to characterize $\mathbb{I}$. Firstly, we notice there is a density operator $\rho' \in \mathcal{D}(\mathbb{I})$ such that

$$\bigvee_{j=1}^{m} \operatorname{supp}(\mathcal{E}_j(\rho')) = \bigvee_{\rho \in \mathcal{D}(\mathbb{I})} \bigvee_{j=1}^{m} \operatorname{supp}(\mathcal{E}_j(\rho)) = \mathbb{I}, \tag{8a}$$

since $\mathbb{I}$ is a linear space of finite dimension and ρ' can be chosen to be a mixture $\sum_k p_k |\psi_k\rangle\langle\psi_k|$ of finitely many pure states $|\psi_k\rangle$, each contributing at least one linearly independent element in $\mathbb{I}$. We further mix ρ' to $\rho'' = \frac{1}{m+1}[\rho' + \sum_{j=1}^{m} \mathcal{E}_j(\rho')]$, so that

$$\mathrm{supp}(\rho'') = \bigvee_{j=1}^{m} \mathrm{supp}(\mathcal{E}_j(\rho'')). \tag{8b}$$

On the other hand, we have

$$\mathrm{supp}(\bar{\mathcal{E}}(\rho'')) = \bigvee_{j=1}^{m} \mathrm{supp}(\mathcal{E}_j(\rho'')), \tag{8c}$$

where $\bar{\mathcal{E}}$ is the arithmetic average of $\mathcal{E}$, i. e. $\bar{\mathcal{E}}(\rho'') = \frac{1}{m}\sum_{j=1}^{m} \mathcal{E}_j(\rho'')$. Combining (8b) and (8c), we get

$$\mathrm{supp}(\rho'') = \mathrm{supp}(\bar{\mathcal{E}}(\rho'')) = \mathbb{I}. \tag{8d}$$

It yields the nice property that the supports of ρ'' and $\bar{\mathcal{E}}(\rho'')$ are both $\mathbb{I}$. We collect those density operators $\rho'' \in \mathcal{D}(\mathbb{I})$ satisfying that property into the set Γ, which is convex and compact in the viewpoint of probabilistic ensemble. Since $\bar{\mathcal{E}}$ is a continuous function from Γ to itself, it follows from Brouwer's fixedpoint theorem [33, Chapter 4] that there exists a fixedpoint $\gamma \in \Gamma$ of $\bar{\mathcal{E}}$ characterized by the stationary equation

$$\bar{\mathcal{E}}(\gamma) = \gamma, \tag{9}$$

where γ is a Hermitian matrix of variables and $\bar{\mathcal{E}}$ gives rise to coefficients. The stationary equation is a system of linear equations that can be efficiently solved. Here we relax the restriction $\gamma \in \Gamma$ to $\gamma \in \mathcal{H}(\mathbf{M_{true}}(\mathbb{H}))$ for the consideration of efficiency, since Hermitian operators are much easier to be encoded than positive ones. How to recover from the restriction $\gamma \in \Gamma$ and even compute the invariant space $\mathbb{I}$ is ensured by the following lemma.

Lemma 7.3 ([61, Lemma 5.4 *&* Algorithm 1]). *Let γ_0 be a nonzero solution of the stationary equation* (9)*. Then* $\mathrm{supp}(\gamma_0)$ *is an invariant space $\mathbb{I}$ of $\mathcal{P}$, which can be computed in time $O(d_0^6)$ with $d_0 = \dim(\mathbf{M_{true}}(\mathbb{H}))$.*

Example 7.4. Consider the while-loop $\mathcal{P}_2$ attached with two nondeterministic super-operators $\mathcal{E}(\alpha_1) = \mathcal{E}_1 = \{\mathbf{H} \otimes \mathbf{X}\}$ and $\mathcal{E}(\alpha_2) = \mathcal{E}_2 = \{\mathbf{X} \otimes \mathbf{H}\}$ in Example 3.7, the average super-operator $\bar{\mathcal{F}} = \{\mathbf{F}_1, \mathbf{F}_2\}$ is given by the Kraus operators

$$\mathbf{F}_1 = \tfrac{1}{\sqrt{2}}(|+,1\rangle\langle 0,0| + |-,1\rangle\langle 1,0| + |-,0\rangle\langle 1,1|),$$
$$\mathbf{F}_2 = \tfrac{1}{\sqrt{2}}(|1,+\rangle\langle 0,0| + |0,+\rangle\langle 1,0| + |0,-\rangle\langle 1,1|).$$

Since the stationary equation $\bar{\mathcal{F}}(\gamma) = \gamma$ with $\gamma \in \mathcal{H}(\mathbf{M_{true}}(\mathbb{H}_{\mathbf{VAR}}))$ has no nonzero solution, $\mathcal{P}_2$ has no invariant space and thus is universally terminating.

To explicitly illustrate the method, we reset $\mathcal{E}(\alpha_1) = \mathcal{E}_1' = \{\mathbf{X} \otimes \mathbf{X}\}$ and $\mathcal{E}(\alpha_2) = \mathcal{E}_2' = \{\mathbf{H} \otimes \mathbf{H}\}$ as the super-operators for the modified while-loop $\mathcal{P}_2'$. We can compute the average super-operator of $\mathcal{P}_2'$ as $\bar{\mathcal{F}}' = \{\mathbf{F}_1', \mathbf{F}_2'\}$ with Kraus operators

$$\mathbf{F}_1' = \tfrac{1}{\sqrt{2}}(|1,1\rangle\langle 0,0| + |0,1\rangle\langle 1,0| + |0,0\rangle\langle 1,1|),$$
$$\mathbf{F}_2' = \tfrac{1}{\sqrt{2}}(|+,+\rangle\langle 0,0| + |-,+\rangle\langle 1,0| + |-,-\rangle\langle 1,1|).$$

Solving the stationary equation $\bar{\mathcal{F}}'(\gamma) = \gamma$ with $\gamma \in \mathcal{H}(\mathbf{M_{true}}(\mathbb{H}_{\mathbf{VAR}}))$, we obtain the unique solution $\gamma_0 = (|0,0\rangle + |1,1\rangle)(\langle 0,0| + \langle 1,1|)$. Hence the invariant space $\mathbb{I}$ of $\mathcal{P}'$ is actually $\mathrm{supp}(\gamma_0) = \mathrm{span}(\{|0,0\rangle + |1,1\rangle\})$, which entails that $\mathcal{P}_2'$ is not universally terminating. □

Whenever the program has no invariant space, every input state has its own scheduler that achieves the termination with probability one. In the following, we are to synthesize a universal scheduler that forces all input states to be terminating with probability one. The procedure of synthesizing such a universal scheduler is stated in Algorithm 5. In detail, each inner loop (Line 3) attempts to find a pure state $|\psi\rangle$ in the orthocomplement $\mathbb{S}^\perp$ of $\mathbb{S}$ that is terminating with positive probability under some finite scheduler ς. It is realizable per outer loop (Line 2), since otherwise

$$\begin{aligned} \neg\exists\ |\psi\rangle \in \mathbb{S}^\perp\ \exists\,\alpha_j \in \Sigma : \operatorname{supp}(\mathcal{E}_j(|\psi\rangle\langle\psi|)) \not\subseteq \mathbb{S} &\iff \forall\ |\psi\rangle \in \mathbb{S}^\perp\ \forall\,\alpha_j \in \Sigma : \operatorname{supp}(\mathcal{E}_j(|\psi\rangle\langle\psi|)) \subseteq \mathbb{S}^\perp \\ &\iff \forall\,\rho \in \mathcal{D}(\mathbb{S}^\perp)\ \forall\,\alpha_j \in \Sigma : \operatorname{supp}(\mathcal{E}_j(\rho)) \subseteq \mathbb{S}^\perp \end{aligned}$$

entailing $\mathbb{S}^\perp$ is invariant, which contradicts the assumption that there is no invariant space. Utilizing this property, we avoid the exponential-up enumerating all finite schedulers with length not greater than d in [71, Algorithm 1] for expanding $\mathbb{S}$ by one dimension or more, which yields the desired polynomial-time efficiency. The correctness of Algorithm 5 is guaranteed by the following lemma, and the complexity is provided below the statements of Algorithm 5.

Lemma 7.5. *If the finite scheduler ς forces all input states to be terminating with positive probability, the infinite scheduler ς^ω forces all input states to be terminating with probability one.*

Proof. For each density operator ρ on $\mathbb{S}^\perp$, we know that it has a positive probability $p(\rho)$ of termination under the finite scheduler ς. Thus there is an open disk $\delta(\rho)$ around ρ, in which each density operator has termination probability at least $\frac{1}{2}p(\rho)$. Since $\mathcal{D}(\mathbb{S}^\perp)$ is a compact set, the open cover $\{\delta(\rho) : \rho \in \mathcal{D}(\mathbb{S}^\perp)\}$ of $\mathcal{D}(\mathbb{S}^\perp)$ has a subcover $\{\delta(\rho_i) : i \in \mathit{IDX}\}$ with some finite index set IDX, i. e. $|\mathit{IDX}| < \infty$. Let $p = \min_{i\in \mathit{IDX}} p(\rho_i)$, which is clearly a nonzero constant. Performing the finite scheduler ς once, all density operators ρ on $\mathbb{S}^\perp$ have termination probability at least $\frac{p}{2}$; performing ς infinitely many times, all density operators ρ on $\mathbb{S}^\perp$ achieve the termination probability one. □

Sample-based Verification. Model checking [13] advocates the verification technology via an exhaustive check over a finite set of system states, namely *samples*. Usually it cannot be applied to systems with continuous sample spaces, e. g. the Hilbert space $\mathbb{H}$ which is a continuum. The above lemma, however, suggests that the exhaustive check still works, when samples to be checked are well elaborately chosen from $\mathcal{D}(\mathbb{S}^\perp)$, such that the set of samples is finite, i. e. $|\mathit{IDX}| < \infty$. Thereby, it can be developed to a practically useful verification technology using samples like [34], which will be popular in modern software engineering. □

Algorithm 5 Synthesizing a Universal Scheduler for Termination

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$ and $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ over $\mathbb{H}$ with dimension d that has no invariant space;

Output: a universal scheduler under which $\mathcal{P}$ terminates with probability one on all input states.

```
1: $\mathbb{S} \leftarrow \mathbf{M}_{\mathbf{false}}(\mathbb{H})$ and $\varsigma \leftarrow \epsilon$;
2: while $\mathbb{S} \neq \mathbb{H}$ do
3:     for all $\alpha_j \in \Sigma$ do
4:         if there is a $|\psi\rangle \in \mathbb{S}^\perp$ such that $\operatorname{supp}(\mathcal{E}_j(|\psi\rangle\langle\psi|))$ is not orthogonal to $\mathbb{S}$ then
5:             let $|\psi\rangle$ be such an element in $\mathbb{S}^\perp$;
6:             $\mathbb{S} \leftarrow \mathbb{S} \vee \operatorname{span}(\{|\psi\rangle\})$ and $\varsigma \leftarrow \varsigma \cdot \alpha_j$;
7:             break;
8: return $\varsigma^\omega$.
```

Complexity. Note that there are less than $m \cdot d$ times of entering the inner loop in Line 3. Each inner loop seeks a pure state $|\psi\rangle \in \mathbb{S}^\perp$, satisfying that $\mathbf{E}_k |\psi\rangle$ is not orthogonal to $\mathbb{S}$ for some Kraus operator $\mathbf{E}_k$ in the Kraus representation $\mathcal{E}_j = \{\mathbf{E}_k : k = 1, 2, \ldots, K\}$ where $K \leq d^2$. Let $\{|\psi_l\rangle : l = 1, 2, \ldots, L\}$ with $L < d$ be the orthonormal basis of $\mathbb{S}$. For a fixed pair of $\mathbf{E}_k$ and $|\psi_l\rangle$, determining whether $\mathbf{E}_k |\psi\rangle$ is orthogonal to $|\psi_l\rangle$ amounts to solving the linear equation $\langle\psi_l| \mathbf{E}_k |\psi\rangle = 0$, which costs $O(d^3)$ operations. Hence Algorithm 5 is in polynomial time $O(K \cdot L \cdot m \cdot d^3) \subseteq O(m \cdot d^6)$. □

Example 7.6. In Example 7.4, we have known that $\mathcal{P}_2$ is universally terminating. By Algorithm 5, we can synthesize a universal scheduler that forces all input states to be terminating with probability one as follows.

(1) Initially, we have $\mathbb{S}_0 = \text{span}(\{|0,1\rangle\})$ and $\varsigma_0 = \epsilon$.

(2) For $\mathbb{S}_0^\perp = \text{span}(\{|0,0\rangle, |1,0\rangle, |1,1\rangle\})$, we can find a pure state $|\psi_1\rangle = |+,0\rangle \in \mathbb{S}_0^\perp$ such that $\text{supp}(\mathcal{E}_1(|\psi_1\rangle\langle\psi_1|)) = \text{span}(\{|0,1\rangle\}) = \mathbb{S}_0$. Then we update

$$\mathbb{S}_1 = \mathbb{S}_0 \vee \text{span}(\{|\psi_1\rangle\}) = \text{span}(\{|0,1\rangle, |+,0\rangle\}) \quad \text{and} \quad \varsigma_1 = \varsigma_0 \cdot \alpha_1 = \alpha_1.$$

(3) Next, for $\mathbb{S}_1^\perp = \text{span}(\{|-,0\rangle, |1,1\rangle\})$, we can find a pure state $|\psi_2\rangle = |1,1\rangle \in \mathbb{S}_1^\perp$ such that $\text{supp}(\mathcal{E}_2(|\psi_2\rangle\langle\psi_2|)) = \text{span}(\{|0,-\rangle\})$ which is not orthogonal to $\mathbb{S}_1$. Then we update

$$\mathbb{S}_2 = \mathbb{S}_1 \vee \text{span}(\{|\psi_2\rangle\}) = \text{span}(\{|0,1\rangle, |+,0\rangle, |1,1\rangle\}) \quad \text{and} \quad \varsigma_2 = \varsigma_1 \cdot \alpha_2 = \alpha_1\alpha_2.$$

(4) Finally, for $\mathbb{S}_2^\perp = \text{span}(\{|-,0\rangle\})$, the pure state $|\psi_3\rangle = |-,0\rangle \in \mathbb{S}_2^\perp$ gives $\text{supp}(\mathcal{E}_1(|\psi_3\rangle\langle\psi_3|)) = \text{span}(\{|1,1\rangle\})$ which is not orthogonal to $\mathbb{S}_2$, and we get

$$\mathbb{S}_3 = \text{span}(\{|0,1\rangle, |+,0\rangle, |1,1\rangle, |-,0\rangle\}) = \mathbb{H}_{\mathbf{VAR}} \quad \text{and} \quad \varsigma_3 = \alpha_1\alpha_2\alpha_1.$$

Thereby, $\varsigma_3 = \alpha_1\alpha_2\alpha_1$ is the finite scheduler that forces all input states to be terminating with positive probability, and ς_3^ω is the infinite scheduler that forces all input states to be terminating with probability one. □

By Lemma 7.3 and Algorithm 5, we obtain the result:

Theorem 7.7. *The universal termination problem described in Problem 3.14 can be solved in polynomial time.*

As an immediate corollary, we get that it is in polynomial time to synthesize a scheduler for the termination if exists.

8 CONCLUSION

In this paper, we have studied the model of nondeterministic quantum program and the termination and the universal termination problems. To decide the termination, we needed two ingredients. One was computing the reachable space of a program fed with an input state, that was a superset of the set of reachable states but was of explicit algebraic structure. A more precise over-approximation of the reachable set was proposed and could be computed in polynomial time. The other was computing the divergent set of a program, which could be obtained in exponential time. The termination follows from the necessary and sufficient condition that the two sets were disjoint.

For the universal termination, the necessary and sufficient condition was the existence of invariant space, which could be detected in polynomial time. Once a program was decided to be universally terminating, a universal scheduler would be synthesized in polynomial time to force all input states to be terminating with probability one. A case study of the quantum Bernoulli factory protocol was provided to demonstrate our methods.

For future work, we would like to:

- consider the weak termination problem, as described in Problem 3.12, over nondeterministic quantum programs;

- synthesize the optimal scheduler that minimizes the expected execution time for a specified input state (resp. all input states with uniform distribution), whenever the termination (resp. universal termination) is guaranteed.

Here, for a specified input state ρ, the expected execution time under an infinite scheduler σ is defined by $\mathrm{TE}_\sigma(\rho) = \sum_{i=0}^{\infty} i \cdot \mathrm{tr}(\mathbf{M}_{\mathbf{true}}\mathcal{F}_{\sigma\uparrow i}(\rho))$; the expected execution time under an optimal infinite scheduler is $\inf_{\sigma\in\Sigma^\omega} \mathrm{TE}_\sigma(\rho)$. When the input state is unspecified, we could choose the input state as the uniform distribution $\rho = \mathbf{I}/d$.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant Nos. 12271172, 11871221, 62072176 & 61832015, the National Key R&D Program of China under Grant No. 2018YFA0306704, the Fundamental Research Funds for the Central Universities under Grant No. 2021JQRH014, and the "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software under Grant No. 22510750100.

REFERENCES

[1] Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. 2018. Lexicographic Ranking Supermartingales: An Efficient Approach to Termination of Probabilistic Programs. *Proceedings of the ACM on Programming Languages* 2, POPL, Article 34 (2018), 32 pages.

[2] Shaukat Ali, Tao Yue, and Rui Abreu. 2022. When Software Engineering Meets Quantum Computing. *Commun. ACM* 65, 4 (2022), 84–88.

[3] Thorsten Altenkirch and Jonathan Grattage. 2005. A Functional Quantum Programming Language. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*. IEEE Computer Society, Washington, 249–258.

[4] Martin Avanzini, Georg Moser, Romain Péchoux, Simon Perdrix, and Vladimir Zamdzhiev. 2022. Quantum Expectation Transformers for Cost Analysis. In *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, Christel Baier and Dana Fisman (Eds.). ACM, New York, Article 10, 13 pages.

[5] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking*. MIT Press, Cambridge, MA.

[6] Gilles Barthe, Thomas Espitau, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. 2018. An Assertion-Based Program Logic for Probabilistic Programs. In *Programming Languages and Systems - 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018 (LNCS, Vol. 10801)*, Amal Ahmed (Ed.). Springer, Berlin, 117–144.

[7] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. 2006. *Algorithms in Real Algebraic Geometry* (2 ed.). Springer, Berlin.

[8] Kevin Batz, Tom J. Biskup, Joost-Pieter Katoen, and Tobias Winkler. 2024. Programmatic Strategy Synthesis: Resolving Nondeterminism in Probabilistic Programs. *Proceedings of the ACM on Programming Languages* 8, POPL (2024), 2792–2820.

[9] Amir M. Ben-Amram and Samir Genaim. 2014. Ranking Functions for Linear-Constraint Loops. *Journal of the ACM* 61, 4, Article 26 (2014), 55 pages.

[10] Aleksandar Chakarov and Sleksandar Sankaranarayanan. 2013. Probabilistic Program Analysis with Martingales. In *Computer Aided Verification, 25th International Conference, CAV 2013 (LNCS, Vol. 8044)*, Natasha Sharygina and Helmut Veith (Eds.). Springer, Berlin, 511–526.

[11] Krishnendu Chatterjee, Hongfei Fu, and Amir K. Goharshady. 2016. Termination Analysis of Probabilistic Programs through Positivstellensatz's. In *Computer Aided Verification, 28th International Conference, CAV 2016, Part I (LNCS, Vol. 9779)*, Swarat Chaudhuri and Azadeh Farzan (Eds.). Springer, Berlin, 3–22.

[12] Isaac L. Chuang, Neil A. Gershenfeld, and Mark Kubinec. 1998. Experimental Implementation of Fast Quantum Searching. *Physical Review Letters* 80 (1998), 3408–3411.

[13] E. M. Clarke, E. A. Emerson, and A. P. Sistla. 1986. Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. *ACM Transactions on Programming Languages and Systems* 8, 2 (1986), 244–263.

[14] Patrick Cousot and Radhia Cousot. 1977. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *4th ACM Symposium on Principles of Programming Languages*, Robert M. Graham, Michael A. Harrison, and Ravi Sethi (Eds.). ACM, New York, 238–252.

[15] C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk. 2017. A Storm is Coming: A Modern Probabilistic Model Checker. In *Computer Aided Verification: 29th International Conference, CAV 2017, Part II (LNCS, Vol. 10427)*, Rupak Majumdar and Viktor Kuncak (Eds.). Springer, Berlin, 592–600.

[16] Ellie D'Hondt and Prakash Panangaden. 2006. Quantum Weakest Preconditions. *Mathematical Structures in Computer Science* 16, 3 (2006), 429–451.

[17] Edsger W. Dijkstra. 1976. *A Discipline of Programming*. Prentice-Hall, New York.

[18] Wang Fang and Mingsheng Ying. 2024. Symbolic Execution for Quantum Error Correction Programs. *Proceedings of the ACM on Programming Languages* 8, PLDI (2024), 1040–1065.

[19] Yuan Feng, Runyao Duan, Zheng-Feng Ji, and Mingsheng Ying. 2007. Proof Rules for the Correctness of Quantum Programs. *Theoretical Computer Science* 386, 1-2 (2007), 151–166.

[20] Yuan Feng and Sanjiang Li. 2023. Abstract Interpretation, Hoare Logic, and Incorrectness Logic for Quantum Programs. *Information and Computation* 294, Article 105077 (2023), 22 pages.

[21] Yuan Feng and Yingte Xu. 2023. Verification of Nondeterministic Quantum Programs. In *28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ASPLOS 2023*, Tor M. Aamodt, Natalie D. Enright Jerger, and Michael M. Swift (Eds.). ACM, New York, 789–805.

[22] Yuan Feng and Mingsheng Ying. 2021. Quantum Hoare Logic with Classical Variables. *ACM Transactions on Quantum Computing* 2, 4, Article 16 (2021), 43 pages.

[23] Yuan Feng, Nengkun Yu, and Mingsheng Ying. 2013. Model Checking Quantum Markov Chains. *J. Comput. System Sci.* 79, 7 (2013), 1181–1198.

[24] Luis M. F. Fioriti and Holger Hermanns. 2015. Probabilistic Termination: Soundness, Completeness, and Compositionality. In *42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015*, Sriram. K. Rajamani and David Walker (Eds.). ACM, New York, 489–501.

[25] Hongfei Fu and Krishnendu Chatterjee. 2019. Termination of Nondeterministic Probabilistic Programs. In *Verification, Model Checking, and Abstract Interpretation - 20th International Conference, VMCAI 2019 (LNCS, Vol. 11388)*, Constantin Enea and Ruzica Piskac (Eds.). Springer, Berlin, 468–490.

[26] Google. 2018. Cirq: A Python Library for Writing, Manipulating, and Optimizing Quantum Circuits and Running Them against Quantum Computers and Simulators. https://github.com/quantumlib/Cirq.

[27] Ji Guan, Yuan Feng, Andrea Turrini, and Mingsheng Ying. 2024. Measurement-Based Verification of Quantum Markov Chains. In *Computer Aided Verification - 36th International Conference, CAV 2024, Part III (LNCS, Vol. 14683)*, Arie Gurfinkel and Vijay Ganesh (Eds.). Springer, Berlin, 533–554.

[28] Jifeng He, Karen Seidel, and Annabelle McIver. 1997. Probabilistic Models for the Guarded Command Language. *Science of Computer Programming* 28, 2–3 (1997), 171–192.

[29] Nils Herrmann, Daanish Arya, Marcus W. Doherty, Angus Mingare, Jason C. Pillay, Florian Preis, and Stefan Prestel. 2023. Quantum Utility — Definition and Assessment of a Practical Quantum Advantage. In *2023 IEEE International Conference on Quantum Software (QSW)*. IEEE Computer Society, Washington, 162–174.

[30] C. A. R. Hoare. 1969. An Axiomatic Basis for Computer Programming. *Commun. ACM* 12, 10 (1969), 576–580.

[31] IBM. 2020. Qiskit: An Open-source SDK for Working with Quantum Computers at the Level of Pulses, Circuits, and Algorithms. https://github.com/QISKit.

[32] IBM. 2023. IBM Debuts Next-Generation Quantum Processor & IBM Quantum System Two, Extends Roadmap to Advance Era of Quantum Utility. https://newsroom.ibm.com/.

[33] Vasile I. Istrăţescu. 2001. *Fixed Point Theory: An Introduction*. Springer, Berlin.

[34] Hui Jiang, Jianling Fu, Ming Xu, Yuxin Deng, and Zhi-Bin Li. 2024. A Sample-Driven Solving Procedure for the Repeated Reachability of Quantum Continuous-time Markov Chains. In *27th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2024*. ACM, New York, Article 9, 12 pages.

[35] Jiaqing Jiang, Jialin Zhang, and Xiaoming Sun. 2018. Quantum-to-Quantum Bernoulli Factory Problem. *Physical Review A* 97, 3, Article 032303 (2018), 6 pages.

[36] Karuna Kadian, Sunita Garhwal, and Ajay Kumar. 2021. Quantum Walk and Its Application Domains: A Systematic Review. *Computer Science Review* 41, Article 100419 (2021), 52 pages.

[37] Benjamin L. Kaminski and Joost-Pieter Katoen. 2015. On the Hardness of Almost-Sure Termination. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Part I (LNCS, Vol. 9234)*, Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella (Eds.). Springer, Berlin, 307–318.

[38] Benjamin L. Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2016. Weakest Precondition Reasoning for Expected Run-Times of Probabilistic Programs. In *Programming Languages and Systems, 25th European Symposium on Programming, ESOP 2016 (LNCS, Vol. 9632)*, Peter Thiemann (Ed.). Springer, Berlin, 364–389.

[39] Benjamin L. Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. 2018. Weakest Precondition Reasoning for Expected Runtimes of Randomized Algorithms. *Journal of the ACM* 65, 5, Article 30 (2018), 68 pages.

[40] Joost-Pieter Katoen, Friedrich Gretz, Nils Jansen, Benjamin L. Kaminski, and Federico Olmedo. 2015. Understanding Probabilistic Programs. In *Correct System Design (LNCS, Vol. 9360)*, Roland Meyer, André Platzer, and Heike Wehrheim (Eds.). Springer, Berlin, 15–32.

[41] M. S. Keane and George L. O'Brien. 1994. A Bernoulli Factory. *ACM Transactions on Modeling and Computer Simulation* 4, 2 (1994), 213–219.

[42] Dexter Kozen. 1979. Semantics of Probabilistic Programs. In *20th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, 101–114.

[43] Dexter Kozen. 1983. A Probabilistic PDL. In *15th Annual ACM Symposium on Theory of Computing*, David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas (Eds.). ACM, New York, 291–297.

[44] M. Kwiatkowska, G. Norman, and D. Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Computer Aided Verification: 23rd International Conference, CAV 2011 (LNCS, Vol. 6806)*, G. Gopalakrishnan and S. Qadeer (Eds.). Springer, Berlin, 585–591.

[45] Marco Lewis, Sadegh Soudjani, and Paolo Zuliani. 2023. Formal Verification of Quantum Programs: Theory, Tools, and Challenges. *ACM Transactions on Quantum Computing* 5, 1, Article 1 (2023), 35 pages.

[46] Yangjia Li and Mingsheng Ying. 2018. Algorithmic Analysis of Termination Problems for Quantum Programs. *Proceedings of the ACM on Programming Languages* 2, POPL, Article 35 (2018), 29 pages.

[47] Yangjia Li, Nengkun Yu, and Mingsheng Ying. 2014. Termination of Nondeterministic Quantum Programs. *Acta Informatica* 51, 1 (2014), 1–24.

[48] Junyi Liu, Bohua Zhan, Shuling Wang, Shenggang Ying, Tao Liu, Yangjia Li, Mingsheng Ying, and Naijun Zhan. 2019. Formal Verification of Quantum Algorithms Using Quantum Hoare Logic. In *Computer Aided Verification - 31st International Conference, CAV 2019, Part II (LNCS, Vol. 11562)*, Isil Dillig and Serdar Tasiran (Eds.). Springer, Berlin, 187–207.

[49] Junyi Liu, Li Zhou, Gilles Barthe, and Mingsheng Ying. 2022. Quantum Weakest Preconditions for Reasoning about Expected Runtimes of Quantum Programs. In *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, Christel Baier and Dana Fisman (Eds.). ACM, New York, Article 4, 13 pages.

[50] Annabelle McIver and Carroll Morgan. 2006. *Abstraction, Refinement and Proof for Probabilistic Systems.* Springer, Berlin.

[51] Annabelle McIver, Carroll Morgan, Benjamin L. Kaminski, and Joost-Pieter Katoen. 2018. A New Proof Rule for Almost-Sure Termination. *Proceedings of the ACM on Programming Languages* 2, POPL, Article 33 (2018), 28 pages.

[52] Marcel Moosbrugger, Ezio Bartocci, Joost-Pieter Katoen, and Laura Kovács. 2021. Automated Termination Analysis of Polynomial Probabilistic Programs. In *Programming Languages and Systems - 30th European Symposium on Programming, ESOP 2021 (LNCS, Vol. 12648)*, Nobuko Yoshida (Ed.). Springer, Berlin, 491–518.

[53] Carroll Morgan, Annabelle McIver, and Karen Seidel. 1996. Probabilistic Predicate Transformers. *ACM Transactions on Programming Languages and Systems* 18, 3 (1996), 325–353.

[54] Markus Müller-Olm and Helmut Seidl. 2004. Computing Polynomial Program Invariants. *Inform. Process. Lett.* 91, 5 (2004), 233–244.

[55] Juan M. Murillo, José García-Alonso, Enrique Moguel, Johanna Barzen, Frank Leymann, Shaukat Ali, Tao Yue, Paolo Arcaini, Ricardo Pérez, Ignacio G. R. de Guzmán, Mario Piattini, Antonio Ruiz-Cortés, Antonio Brogi, Jianjun Zhao, Andriy V. Miranskyy, and Manuel Wimmer. 2024. Challenges of Quantum Software Engineering for the Next Decade: The Road Ahead. *CoRR* abs/2404.06825 (2024), 12 pages. https://doi.org/10.48550/arXiv.2404.06825

[56] Michael A. Nielsen and Isaac L. Chuang. 2000. *Quantum Computation and Quantum Information.* Cambridge University Press, London.

[57] Bernhard Ömer. 1998. *A Procedural Formalism for Quantum Computing.* Technical Report. Technical University of Vienna.

[58] Jeff W. Sanders and Paolo Zuliani. 2000. Quantum Programming. In *Mathematics of Program Construction*, R. Backhouse and J. N. Oliveira (Eds.). Springer, Berlin, 80–99.

[59] Peter Selinger. 2004. Towards a Quantum Programming Language. *Mathematical Structures in Computer Science* 14, 4 (2004), 527–586.

[60] Krysta Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. 2018. Q#: Enabling Scalable Quantum Computing and Development with a High-level DSL. In *Proceedings of the Real World Domain Specific Languages Workshop, RWDSL@CGO 2018.* ACM, New York, Article 7, 10 pages.

[61] Ming Xu, Jianling Fu, Jingyi Mei, and Yuxin Deng. 2022. An Algebraic Method to Fidelity-Oriented Model Checking over Quantum Markov Chains. *Theoretical Computer Science* 935 (2022), 61–81.

[62] Ming Xu, Jianling Fu, Jingyi Mei, and Yuxin Deng. 2022. Model Checking QCTL Plus on Quantum Markov Chains. *Theoretical Computer Science* 913 (2022), 43–72.

[63] Ming Xu, Cheng-Chao Huang, and Yuan Feng. 2021. Measuring the Constrained Reachability in Quantum Markov Chains. *Acta Informatica* 58, 6 (2021), 653–674.

[64] Mingsheng Ying. 2011. Floyd–Hoare Logic for Quantum Programs. *ACM Transactions on Programming Languages and Systems* 33, 6, Article 19 (2011), 49 pages.

[65] Mingsheng Ying. 2016. *Foundations of Quantum Programming.* Morgan Kaufmann, Amsterdam.

[66] Mingsheng Ying. 2019. Toward Automatic Verification of Quantum Programs. *Formal Aspects of Computing* 31, 1 (2019), 3–25.

[67] Mingsheng Ying, Runyao Duan, Yuan Feng, and Zhengfeng Ji. 2010. Predicate Transformer Semantics of Quantum Programs. *Semantic Techniques in Quantum Computation* 8 (2010), 311–360.

[68] Mingsheng Ying and Yuan Feng. 2010. Quantum Loop Programs. *Acta Informatica* 47, 4 (2010), 221–250.

[69] Mingsheng Ying, Shenggang Ying, and Xiaodi Wu. 2017. Invariants of Quantum Programs: Characterisations and Generation. In *44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017*, Giuseppe Castagna and Andrew D. Gordon (Eds.). ACM,

New York, 818–832.

[70] Mingsheng Ying, Nengkun Yu, Yuan Feng, and Runyao Duan. 2013. Verification of Quantum Programs. *Science of Computer Programming* 78, 9 (2013), 1679–1700.

[71] Shenggang Ying and Mingsheng Ying. 2018. Reachability Analysis of Quantum Markov Decision Processes. *Information and Computation* 263 (2018), 31–51.

[72] Nengkun Yu and Jens Palsberg. 2021. Quantum Abstract Interpretation. In *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, Stephen N. Freund and Eran Yahav (Eds.). ACM, New York, 542–558.

[73] Nengkun Yu and Mingsheng Ying. 2012. Reachability and Termination Analysis of Concurrent Quantum Programs. In *CONCUR 2012 - Concurrency Theory - 23rd International Conference (LNCS, Vol. 7454)*, Maciej Koutny and Irek Ulidowski (Eds.). Springer, Berlin, 69–83.

[74] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. 2020. Quantum Computational Advantage Using Photons. *Science* 370, 6523 (2020), 1460–1463.

[75] Paolo Zuliani. 2004. Non-deterministic Quantum Programming. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages (TUCS General Publication, 33)*. Turku Centre for Computer Science, Turku, 179–195.

A EXERCISE 2.73 IN REFERENCE 38

Let ρ be a density operator. A *minimal ensemble* for ρ is an ensemble $\{(p_k, |\psi_k\rangle) : k = 1, 2, \ldots, K\}$ containing a number K of elements equal to the rank of ρ. Let $|\psi\rangle$ be any state in the support of ρ. Show that there is a minimal ensemble for ρ that contains $|\psi\rangle$, and moreover that in any such ensemble $|\psi\rangle$ must appear with a constructive probability.

Proof. Let $\sum_{k=1}^{K} p_k |\psi_k\rangle\langle\psi_k|$ be the spectral decomposition of ρ, where p_k are all nonzero eigenvalues of ρ, $|\psi_k\rangle$ are the corresponding eigenvectors, and K is the rank of ρ. Clearly, the support Ξ of ρ is $\mathrm{span}(\{|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_K\rangle\})$. For any $|\psi\rangle \in \Xi$, by the orthonormality of eigenvectors, we can uniquely determine the subspace $\Xi' \subseteq \Xi$ that is spanned by those $|\psi_k\rangle$ satisfying $\langle\psi_k|\psi\rangle \neq 0$. Namely, this Ξ' is spanned by $|\psi_{k'}\rangle$. We claim there is a constructive positive value of p, such that $\det_{\Xi'}(\rho - p|\psi\rangle\langle\psi|) = 0$, where $\det_{\Xi'}$ is the determinant on the subspace Ξ' of $\mathbb{H}$. Then, letting p_0 be the smallest value of p, we obtain $\rho = p_0|\psi\rangle\langle\psi| + \rho'$, where ρ' is a positive operator with less rank than ρ. Under the spectral decomposition $\sum_{j=1}^{K-1} q_j|\varphi_j\rangle\langle\varphi_j|$, we get the minimal ensemble $\{(p_0, |\psi\rangle)\} \cup \{(q_j, |\varphi_j\rangle) : j = 1, 2, \ldots, K-1\}$ of ρ as desired.

Now we turn to prove the aforementioned claim. Utilizing the facts:

- both ρ and $|\psi\rangle\langle\psi|$ are positive operators, and
- the difference $\rho - p|\psi\rangle\langle\psi|$ would be no longer positive when p is sufficiently large,

the existence of p_0 follows by the middle-value theorem, since $f(p) = \det_{\Xi'}(\rho - p|\psi\rangle\langle\psi|)$ is a continuous function in p, satisfying both $f(0) > 0$ and $\lim_{p\to\infty} f(p) < 0$. Let $\mathbf{U} = \sum_{k'} |k'\rangle\langle\psi_{k'}|/\sqrt{p_{k'}}$. It is easy to see $\mathbf{U}\rho\mathbf{U}^\dagger = \sum_{k'} |\psi_{k'}\rangle\langle\psi_{k'}| = \mathbf{I}_{\Xi'} = \sum_{k'} |k'\rangle\langle k'|$ and $\mathbf{U}|\psi\rangle\langle\psi|\mathbf{U}^\dagger = \sum_{k'} |\langle\psi_{k'}|\psi\rangle|^2 |k'\rangle\langle k'|/p_{k'}$. Furthermore, such values of p should satisfy the following equations:

$$\begin{aligned}
\det_{\Xi'}(\rho - p|\psi\rangle\langle\psi|) = 0 &\iff \det_{\Xi'}(\mathbf{U}(\rho - p|\psi\rangle\langle\psi|)\mathbf{U}^\dagger) = 0 \\
&\iff \det_{\Xi'}\left[\left(\sum_{k'} |k'\rangle\langle k'|\right) - \left(\sum_{k'} \frac{p}{p_{k'}} |\langle\psi_{k'}|\psi\rangle|^2 |k'\rangle\langle k'|\right)\right] = 0 \\
&\iff \prod_{k'}\left(p - \frac{p_{k'}}{|\langle\psi_{k'}|\psi\rangle|^2}\right) = 0.
\end{aligned}$$

It entails that p_0 should be chosen as

$$\min_{k'}\left\{\frac{p_{k'}}{|\langle\psi_{k'}|\psi\rangle|^2}\right\}. \qquad \square$$

B IMPLEMENTATION

Algorithms 1 through 5 have been implemented in the Wolfram language on Mathematica 11.3 with an Intel Core i5-4590 CPU at 3.30GHz. An integrated source file is also available at https://www.wolframcloud.com/obj/tpnqp/Published/TPNQP. We list below the main functions for analyzing the termination and the universal termination problems of nondeterministic quantum programs.

- `ReachableSpaceI`(*Operas_*, *Meas_*, *Inistate_*, *Dims_*) computes the I-reachable subspace w. r. t. an input state and returns an orthonormal basis of that subspace.
- `ReachableSpaceII`(*Operas_*, *Meas_*, *Inibasis_*, *Inistate_*, *Dims_*) computes the II-reachable subspace w. r. t. an input state and returns a linearly independent basis of that subspace.
- `Divergent`(*Operas_*, *Meas_*, *Dims_*, *Sigma_*) computes the set of finite schedulers and the union of their corresponding generated pure divergent sets from which the program has a divergence scheduler, i. e., the program terminates with probability zero.
- `NTScheduler`(*Operas_*, *Meas_*, *Inibasis_*, *Dims_*, *RSI_*, *PD_*, *Sigma_*, *pdAss_*, *actionAss_*) computes a nontermination scheduler under which the program does not terminate with probability one on the input state, once the intersection of the I-reachable space and the pure divergent set is checked to be not empty.
- `UniScheduler`(*Operas_*, *Meas_*, *Dims_*) computes a universal scheduler under which the program terminates with probability one on all input states.

After specifying the Hilbert space, a nondeterministic quantum program and an input state, the five algorithms can be performed by calling these functions.

B.1 Quantum Bernoulli Factory Protocol

For the nondeterministic quantum program describing the quantum Bernoulli factory (QBF) protocol in Example 3.2, by invoking the implemented algorithms, we have validated the nontermination and the universal termination of the program. The detailed performance of the five algorithms is shown in Table 4.

Table 4. Algorithmic performance on the QBF protocol

QBF	ReachSpace-I	ReachSpace-II	Divergent	NTScheduler	UniScheduler
Example No.	4.6	4.10	5.4	6.2	7.6
Time (s)	0.016	0.015	1.422	0.203	2.593
Memory (MB)	132.605	205.833	181.868	133.217	205.833

B.2 Nondeterministic Quantum Walk

Here we consider another example, a quantum walk along a ring with three vertexes in a 3-dimensional Hilbert space [36]. The vertex set is $V = \{|0\rangle, |1\rangle, |2\rangle\}$ entailing $\mathbb{H} = \text{span}(V)$, where $|0\rangle$ denotes the starting position and $|2\rangle$ denotes the absorbing boundary. A projective measurement $\{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\}$ with $\mathbf{M}_{\mathbf{false}} = |2\rangle\langle 2|$ and $\mathbf{M}_{\mathbf{true}} = \mathbf{I} - \mathbf{M}_{\mathbf{false}} = |0\rangle\langle 0| + |1\rangle\langle 1|$ is designed to observe whether the particle is trapped in the boundary after each move. Each move of the particle is modelled by a quantum operation which is nondeterministically chosen from actions $\{w_1, w_2\}$, so that $\mathcal{E}(w_1) = \{\mathbf{W}_1\}$ and $\mathcal{E}(w_2) = \{\mathbf{W}_2\}$ with

$$\mathbf{W}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \varpi & \varpi^2 \\ 1 & \varpi^2 & \varpi \end{bmatrix} \quad \text{and} \quad \mathbf{W}_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \varpi^2 & \varpi \\ 1 & \varpi & \varpi^2 \end{bmatrix}$$

where $\varpi = \mathrm{e}^{2\pi\mathrm{i}/3}$ is the root of unit. Then we can formally describe the quantum walk with nondeterministic moves as the program $\mathcal{P}_3 = (\{w_1, w_2\}, \mathcal{E}, \{\mathbf{M}_{\mathbf{true}}, \mathbf{M}_{\mathbf{false}}\})$.

By invoking the implemented algorithms, we can obtain the following results.

- Starting from position $|0\rangle$, the I-reachable space $\Phi(\mathcal{P}_3, |0\rangle\langle 0|)$ of the particle is
$$\mathrm{span}(\{|0\rangle, (|1\rangle + |2\rangle)/\sqrt{2}, \mathrm{i}(|1\rangle - |2\rangle)/\sqrt{2}\}) = \mathbb{H},$$
and the II-reachable space $\Upsilon(\mathcal{P}_3, |0\rangle\langle 0|)$ is
$$\mathrm{span}\left(\left\{\begin{array}{l}\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \begin{bmatrix}8&2+2\sqrt{3}\mathrm{i}&2-2\sqrt{3}\mathrm{i}\\2-2\sqrt{3}\mathrm{i}&2&-1-\sqrt{3}\mathrm{i}\\2+2\sqrt{3}\mathrm{i}&-1+\sqrt{3}\mathrm{i}&2\end{bmatrix}, \begin{bmatrix}8&2-2\sqrt{3}\mathrm{i}&2+2\sqrt{3}\mathrm{i}\\2+2\sqrt{3}\mathrm{i}&2&-1+\sqrt{3}\mathrm{i}\\2-2\sqrt{3}\mathrm{i}&-1-\sqrt{3}\mathrm{i}&2\end{bmatrix},\\ \begin{bmatrix}14&5-\sqrt{3}\mathrm{i}&11-5\sqrt{3}\mathrm{i}\\5+\sqrt{3}\mathrm{i}&2&5-\sqrt{3}\mathrm{i}\\11+5\sqrt{3}\mathrm{i}&5+\sqrt{3}\mathrm{i}&14\end{bmatrix}, \begin{bmatrix}14&11-5\sqrt{3}\mathrm{i}&5-\sqrt{3}\mathrm{i}\\11+5\sqrt{3}\mathrm{i}&14&5+\sqrt{3}\mathrm{i}\\5+\sqrt{3}\mathrm{i}&5-\sqrt{3}\mathrm{i}&2\end{bmatrix},\\ \begin{bmatrix}26&17+5\sqrt{3}\mathrm{i}&14-2\sqrt{3}\mathrm{i}\\17-5\sqrt{3}\mathrm{i}&14&8-4\sqrt{3}\mathrm{i}\\14+2\sqrt{3}\mathrm{i}&8+4\sqrt{3}\mathrm{i}&8\end{bmatrix}, \begin{bmatrix}26&14-2\sqrt{3}\mathrm{i}&17+5\sqrt{3}\mathrm{i}\\14+2\sqrt{3}\mathrm{i}&8&8+4\sqrt{3}\mathrm{i}\\17-5\sqrt{3}\mathrm{i}&8-4\sqrt{3}\mathrm{i}&14\end{bmatrix}\end{array}\right\}\right).$$
- The divergent set is $PD(\mathcal{P}_3) = \emptyset$, implying no nontermination scheduler, thus the particle is proven to be surely absorbed no matter which move it takes in each step.
- There exists a universal scheduler w_1^{ω} that forces the particle to reach the absorbing boundary regardless of its initial position.

The detailed performance of the five algorithms is shown in Table 5.

Table 5. Algorithmic performance on the nondeterministic quantum walk (NQW)

NQW	ReachSpace-I	ReachSpace-II	Divergent	NTScheduler	UniScheduler
Time (s)	0.032	1.766	0.203	0.000	0.015
Memory (MB)	94.770	205.833	98.328	100.629	176.579

Generally speaking, all of the functions involved in Algorithms 1, 2 and 5 are much efficient as their theoretical complexity has an upper bound of **PTIME**. Those in Algorithms 3 and 4 may be inefficient in the worst case, due to the fact that the derivation tree construction for the pure divergent set is **EXPTIME**. Fortunately, their running time is rather acceptable in our case studies.