Termination Analysis of Nondeterministic Quantum Programs Revisited

Jianling Fu, Hui Jiang, Ming Xu, Yuxin Deng, and Zhi-Bin Li

Shanghai Key Laboratory of Trustworthy Computing, MoE Engineering Research Center of Software/Hardware Co-design Technology and Application, East China Normal University, Shanghai, China

Abstract. Verifying quantum programs has attracted a lot of interest in recent years. In this paper, we consider the termination problem of quantum programs with nondeterminism. To analyze termination effectively, we over-approximate the reachable set of quantum program states by the reachable subspace, which has an explicit algebraic structure. Compared with the counterpart in existing literature, our reachable subspace is more precise and can be computed in polynomial time. We illustrate the algebraic method via a running example — the quantum Bernoulli factory protocol. Moreover, we study the set of divergent states from which the program terminates with probability zero under some scheduler. By exploiting the algebraic structure of the divergent set, we develop an effective approach using the existential theory of the reals. The complexity is shown, for the first time, to be in exponential time.

Keywords: Quantum program · Markov decision process · Termination

1 Introduction

In the field of quantum computing, physical devices have been rapidly developed in the last decades, particularly in very recent years. In October 2019, Google officially announced that its 53-qubit Sycamore processor took about 200 seconds to sample one instance of a quantum circuit that would have taken the world's most powerful supercomputer 10,000 years [4]. Just one year later, the quantum computer Jiuzhang reached quantum supremacy by implementing a type of Boson sampling on 76 photons, in which case the quantum computer spent less than 20 seconds while a classical supercomputer would require 600 million years [45].

Equally important is quantum software, which is crucial in harnessing the power of quantum computers, such as Shor's algorithm with an exponential-level speed-up for integer factorization [37] and Grover's algorithm with a square-level speed-up for unstructured search [17]. The first practical quantum programming language QCL appeared in Ömer's work [31]. The quantum guarded command language (qGCL) was presented to program a "universal" quantum computer [34]. Selinger [36] and Grattage *et al.* [2] respectively proposed functional programming languages QFC and QML with high-level features. Nowadays, several quantum programming languages, e.g., Qiskit [20], Q# [38], Cirq

[15], PyQuil [33], have been proposed for real-world applications. Detailed survey on programming languages can be found in [35,13,16]. Correspondingly, it is necessary to develop verification methods for quantum programs. To this end, one can decompose "total correctness" into "partial correctness" plus "termination" [19]. Hence termination analysis plays a central role in program verification.

In this paper, we focus on nondeterministic quantum programs in finite-dimensional Hilbert spaces, and study the universal termination problem that is a decision problem asking whether a program fed with an input state terminates with probability one under all schedulers. We first give two models of nondeterministic quantum programs: one has finitely many program locations so that it is easier to model practical scenaries, and the other has exactly one. We show that they are of the same expressiveness, and thus adopt the latter for ease of verification. Then, we consider two characterizations of reachable spaces that over-approximate the set of reachable states. The I-reachable space has the type of a subspace of the Hilbert space, as proposed in the literature [24]; and the II-reachable space has the type of a subspace of Hermitian operators on the Hilbert space. Both are computable in polynomial time, but the latter is more precise, as validated by the running example — the quantum Bernoulli factory protocol. Moreover, we study the set of divergent 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 also developed using the existential theory of the reals. The complexity is shown to be in exponential time. Combining the reachable spaces and the divergent set, our termination analysis is completed by checking the disjointness of them.

The main contributions of the current paper are summarized as follows:

- We propose a more precise characterization of reachable space, which can be computed in polynomial time.
- We analyze the complexity of computing the set of divergent states for the first time, thus settling an open problem.
- A case study on the quantum Bernoulli factory protocol is provided to demonstrate our method.

1.1 Related Work

Verification on probabilistic programs Probabilistic programs have several syntactic constructors — probabilistic choice, nondeterministic choice and observation. The termination problem 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?

Fioriti and Hermanns proposed a framework to prove almost-sure termination by *ranking super-martingales* [11], which is analogous to ranking functions on deterministic programs. Chakarov and Sankaranarayanan applied constraint-based techniques to generate linear ranking super-martingales [6]. Chatterjee *et al.* constructed polynomial ranking super-martingales through positivstellensatz's [7]. A polynomial-time procedure was given to synthesize lexicographic ranking super-martingales for linear probabilistic programs [1]. Fu and Chatterjee applied ranking super-martingales to study the positive almost-sure termination of nondeterministic probabilistic programs [12]. McIver and Morgan generalized the *weakest preconditions* of Dijkstra (an approach to prove total correctness) to the *weakest pre-expectations* [27] for analyzing properties of probabilistic guarded command language (pGCL) [18] and for establishing almost-sure termination [26]. Kaminski *et al.* presented a calculus of weakest pre-expectation style for obtaining bounds on the expected running time of probabilistic programs [21]. Verification tools like Amber [28] have been released to automatically prove almost-sure and positive almost-sure termination. However, in the setting of quantum computing, a program state is no longer simply a probabilistic distribution over program variables; it is instead a density operator (positive semi-definite matrix with unit trace) on Hilbert space, which would be further considered in the following.

Verification on quantum programs In 2010, Ying and Feng initialized the verification of quantum loop programs [43] by giving some necessary and sufficient conditions to ensure termination and almost-sure termination. Later on, the classical Floyd–Hoare logic was extended in the quantum setting to be quantum Floyd–Hoare logic [42], and the Sharir–Pnueli–Hart method was also extended from probabilistic programs to quantum programs [41] toward automatic verification [40]. Yu *et al.* considered concurrent quantum programs [44], and reduced the termination problem to the reachability problem of quantum Markov chains [9]. Li *et al.* dealt with nondeterministic quantum programs [24], and proposed the methods for computing the reachable space from an input state, a superset of the set of reachable states, in polynomial time; and the set of divergent states in an effective procedure with unknown complexity. When the two sets are disjoint, the termination of a program can be safely inferred. However, two remaining issues could be addressed, as considered in the present paper, namely, i) how to characterize the reachable space more precisely and ii) how to analyze the complexity of computing the divergent set. Recently, using semi-definite programming, linear ranking super-martingales have been synthesized for quantum programs with nondeterministic choices, namely angelic and demonic choices [23]. There are also some works for verifying various kinds of quantum protocols and quantum algorithms [14,39,3,10,25,32].

Organization The rest of this paper is organized as follows. Section 2 recalls some basic notions and notations from quantum computing. The models of nondeterministic quantum program are introduced in Section 3 together with its termination problems. Then, we compute the reachable spaces and the divergent set respectively in Sections 4 & 5. Combining them, we are able to analyze the termination. Finally, we conclude this paper in Section 6.

2 Preliminaries

Let $\mathbb{H}$ be a Hilbert space with finite dimension d throughout this paper. Here, we recall the Dirac notations that are standard in quantum computing. Interested readers can refer to [30] 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 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 $|\psi\rangle$, interpreted as a *state*, of $\mathbb{H}$ can be expressed 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 $\sum_{i=1}^{d} |c_i|^2 = 1$. The state space of composite quantum system is the product of state spaces. For two subspaces $\mathbb{B}$ and $\mathbb{B}'$, the joint $\mathbb{B} \vee \mathbb{B}'$ is the subspace spanned by the elements of $\mathbb{B}$ and $\mathbb{B}'$, i.e. $\mathrm{span}(\mathbb{B} \cup \mathbb{B}')$.

Let γ be a linear operator on $\mathbb{H}$. It is *Hermitian*, denoted by $\gamma \in \mathcal{H}(\mathbb{H})$, if $\gamma = \gamma^\dagger$. 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 $|\lambda_i\rangle$ are the corresponding eigenvectors. The *support* of γ is the subspace of $\mathbb{H}$ spanned by all eigenvectors associated with nonzero eigenvalues, i.e., $\mathrm{supp}(\gamma) := \mathrm{span}(\{|\lambda_i\rangle : i = 1, 2, \ldots, d \wedge \lambda_i \neq 0\})$. 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 the eigenvalues of $\mathbf{P}$ are 0 and 1.

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 $\{|\psi_i\rangle : i = 1, 2, \ldots, d\}$. A *density* operator ρ, denoted by $\rho \in \mathcal{D}$, is a positive operator with unit trace. A partial density operator ρ, denoted by $\rho \in \mathcal{D}^{\leq 1}$, is a positive operator with trace not greater than 1. 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. The density operators are usually used to describe quantum states. It means that the quantum system is in state $|\lambda_i\rangle$ with probability p_i. When $m = 1$, we know that the system must be in state $|\lambda_1\rangle$ (with probability 1), which is the so-called *pure* state; and otherwise the state is *mixed*.

A super-operator $\mathcal{E}$, denoted by $\mathcal{E} \in \mathcal{S}$, is a linear operator on linear operators. Any quantum operation can be characterized by the (completely-positive) super-operators in the Kraus representation $\mathcal{E} = \{\mathbf{E}_i : 1, 2, \ldots, m\}$: for a given density operator ρ, we have $\mathcal{E}(\rho) = \sum_{i=1}^{m} \mathbf{E}_i \rho \mathbf{E}_i^\dagger$ where the number of Kraus operators $\mathbf{E}_i$ can be bounded by d^2. For two super-operators $\mathcal{E} = \{\mathbf{E}_i : 1, 2, \ldots, m\}$ and $\mathcal{E}' = \{\mathbf{E}'_i : 1, 2, \ldots, m'\}$, the Kraus representation of their sum $\mathcal{E} + \mathcal{E}'$ is $\{\mathbf{E}_i : 1, 2, \ldots, m\} \cup \{\mathbf{E}'_i : 1, 2, \ldots, m'\}$, and that of their composition $\mathcal{E} \circ \mathcal{E}'$ is $\{\mathbf{E}_i\mathbf{E}'_j :$

$1,2,\ldots,m \wedge j=1,2,\ldots,m'\}$. A super-operator $\mathcal{E}$ is *trace-preserving*, denoted by $\mathcal{E} \in \mathcal{S}^{\eqsim\mathcal{I}}$, if $\sum_{i=1}^{m} \mathbf{E}_i^{\dagger}\mathbf{E}_i = \mathbf{I}$; and it is *trace-nonincreasing*, denoted by $\mathcal{E} \in \mathcal{S}^{\lesssim\mathcal{I}}$, if $\mathbf{I} - \sum_{i=1}^{m} \mathbf{E}_i^{\dagger}\mathbf{E}_i$ is positive. Clearly, $\mathcal{E} \in \mathcal{S}^{\eqsim\mathcal{I}}$ means both $\mathcal{E} \in \mathcal{S}^{\lesssim\mathcal{I}}$ and $\mathcal{E} \in \mathcal{S}^{\gtrsim\mathcal{I}}$.

A set of projector $\mathbf{P}_i$ with $i \in I$ forms a *projective measurement* if $\sum_{i\in I} \mathbf{P}_i = \mathbf{I}$. The measurement aims to get information from quantum states, but it also destroys the quantum state. For example, given a quantum state ρ, after the above projective measurement, we will get an index $i \in I$ with probability $p_i = \mathrm{tr}(\mathbf{P}_i\rho)$; when the outcome is i, the final state would be $\mathbf{P}_i\rho\mathbf{P}_i/p_i$.

3 Program Model

In this section, we introduce the two models of nondeterministic quantum programs. The former is more complicated but easier to model practical scenarios while the latter is simpler and thus easier to be verified. They will be shown to have the same expressiveness. So, for ease of verification, we would like to adopt the latter. Based on that, we will propose the termination problem considered in the present paper.

Definition 1. *A nondeterministic quantum program $\mathcal{P}$ on quantum state space $\mathbb{H}$ is a quadruple $(S, \Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$, where*

- $S = \{s_i : i = 1,2,\ldots,n\}$ *is a finite set of (program) locations;*
- $\Sigma = \{\alpha_j : j = 1,2,\ldots,m\}$ *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 $\mathbb{H}$ from location 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 $\alpha_j \in \Sigma$;*
- $\{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\}$ *is a projective measurement on $\mathbb{H}_{\mathrm{cq}} = \mathcal{C} \otimes \mathbb{H}$ with $\mathcal{C} = \mathrm{span}(\{|s_i\rangle : i = 1,2,\ldots,n\})$, and outcomes* t *and* nt *refer to the termination and the nontermination, respectively.*

Note that the program $\mathcal{P}$ has finitely many actions $\alpha_1, \alpha_2, \ldots, \alpha_m$ to choose at each location s_i. Each action α_j ($j \in \{1,2,\ldots,m\}$) is attached by a series of super-operators $\mathcal{E}_{i,j,k}$ with s_k ranging over S. Let us see how a program is executed at a single step.

1. Once the program is executed at each location s_i, the termination measurement $\{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\}$ is firstly applied on the current quantum state ρ_i that is a density operator on $\mathbb{H}_{\mathrm{cq}}$, globally on the superposition $\rho = \sum_{s_i\in S} \rho_i$. If the result is **t**, it forces the program to terminate with the final state $\mathbf{M}_{\mathrm{t}}\rho\mathbf{M}_{\mathrm{t}}/p_{\mathrm{t}}$ where $p_{\mathrm{t}} = \mathrm{tr}(\mathbf{M}_{\mathrm{t}}\rho)$ is the termination probability. On the contrary, if the result is **nt**, it refers to the nontermination with the final state $\mathbf{M}_{\mathrm{nt}}\rho\mathbf{M}_{\mathrm{nt}}/p_{\mathrm{nt}}$ where $p_{\mathrm{nt}} = \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\rho)$ is the nontermination probability. As $\{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\}$ is a projective measurement, we have $p_{\mathrm{t}} + p_{\mathrm{nt}} = \mathrm{tr}(\rho)$.
2. If the program does not terminate, we encode the state $\mathbf{M}_{\mathrm{nt}}\rho_i\mathbf{M}_{\mathrm{nt}}/p_{\mathrm{nt}}$ with probability p_{nt} simply by $\mathbf{M}_{\mathrm{nt}}\rho_i\mathbf{M}_{\mathrm{nt}}$. Then an action α_j is nondeterministically chosen from the action set Σ and the corresponding super-operators

$\mathcal{E}_{i,j,k}$ are performed on the quantum state after measurement. Finally the control location s_i transfers to s_k, the quantum states become $\rho' = \sum_{s_i,s_k \in S} \{|s_k\rangle\langle s_i|\} \otimes \mathcal{E}_{i,j,k}(\mathbf{M}_{\mathrm{nt}}\rho_i\mathbf{M}_{\mathrm{nt}})$, and the program execution goes on.

Thus the nondeterminism in program execution is resolved by fixing a sequence of actions. An infinite sequence $\sigma = \alpha_1\alpha_2\alpha_3 \cdots \in \Sigma^\omega$ is called an *infinite scheduler*; and a finite sequence $\varsigma = \alpha_1\alpha_2 \cdots \alpha_k \in \Sigma^*$ is a *finite scheduler*.

Sometimes, we would consider the program model with only one (program) location, i.e. $S = \{s\}$. Then the program model would become:

Definition 2 ([24, Definition 1]). *A nondeterministic quantum program $\mathcal{P}$ on quantum state space $\mathbb{H}$ is a triple $(\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$, where*

- $\Sigma = \{\alpha_j : j = 1, 2, \ldots, m\}$ *is a finite set of actions;*
- $\mathcal{E} : \Sigma \to \mathcal{S}^{\approx\mathcal{I}}$ *gives rise to the super-operators $\mathcal{E}_j$ on $\mathbb{H}$ by taking action α_j;*
- $\{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\}$ *is a projective measurement on $\mathbb{H}$, which is the same as in Definition 1.*

A single execution step of the program is similar to that defined in Definition 1. Before taking the action, a measurement is performed on the current quantum state to determine whether the program terminates or not. 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 and step like this until it terminates, but it is viewed as staying at the constant location after executing every step.

Although the model in Definition 1 seems much easier to manipulate than that in Definition 2, the two models have the same expressiveness:

- Given a model in Definition 2, we can obtain a model in Definition 1 by setting the singleton location set $S = \{s\}$ and add the constant location information in the super-operators $\mathcal{E}$.
- Conversely, given a model in Definition 1, we can construct a model $(\Sigma, \mathcal{E}', \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ in Definition 2 by
  - enlarging the quantum state space as $\mathbb{H}_{\mathrm{cq}}$; and
  - setting $\mathcal{E}'(\alpha_j) = \sum_{s_i,s_k \in S} \{|s_k\rangle\langle s_i|\} \otimes \mathcal{E}_{i,j,k}$ for each $\alpha_j \in \Sigma$ as a super-operator on $\mathbb{H}_{\mathrm{cq}}$.

Hence, we can freely choose one of the two definitions for convenience. In this paper, we will adopt the model in Definition 2 for ease of verification.

An execution scheduler of a program defined in Definition 2 can be represented as a sequence of actions above. We define the super-operator $\mathcal{F}_{\alpha_i} = \mathcal{E}_i \circ \{\mathbf{M}_{\mathrm{nt}}\}$ $(\alpha_i \in \Sigma)$ as the composite quantum operation upon nontermination measure outcome; let $\varsigma \uparrow k$ be the finite prefix of ς with length k for $k \leq |\varsigma|$, and $\varsigma \downarrow k$ the suffix obtained by removing the k-prefix from ς. Then we have the following inductive construction of the super-operator over a sequence of actions

$$\mathcal{F}_\varsigma = \begin{cases} \mathcal{I} & \text{if } |\varsigma| = 0 \\ \mathcal{F}_{\varsigma\downarrow 1} \circ \mathcal{F}_{\varsigma\uparrow 1} & \text{if } |\varsigma| \geq 1. \end{cases}$$

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 could be extended to infinite schedulers σ.

Example 1. We will study the quantum Bernoulli factory protocol [22] as a running example of our method. The protocol can model Alice and Bob's electing a leader by coin-tossing. Coins are possibly biased. To overcome it, they may adopt the method that:

1. use two coins, which are referred to as the left and the right ones,
2. nondeterministically choose one of them to toss, and
3. meanwhile turn the other over.

If the left coin is head and the right is tail, then Alice wins; if the right coin is head and the left is tail, then Bob wins; and otherwise it tells nothing, they restart the process. Before adopting this election method, Alice and Bob want to know whether the method ensures the fairness that Alice eventually has the chance of winning, as well as Bob. Let us check the former, the latter is similar.

In order to describe the protocol, we design a nondeterministic quantum program as follows. Let $\mathbb{H}$ be the one-qubit Hilbert space with orthonormal basis $\{|0\rangle, |1\rangle\}$ where $|0\rangle$ and $|1\rangle$ denote "head" and "tail" respectively, and $\mathbb{H}^{\otimes 2} := \mathbb{H} \otimes \mathbb{H}$ the two-qubit Hilbert space. It starts with a quantum state $|q_1, q_2\rangle$ in $\mathbb{H}^{\otimes 2}$ to denote the initial state of two individual coins. Tossing a coin is modelled by applying the Hadamard gate $H = |+\rangle\langle 0| + |-\rangle\langle 1|$ with $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$, and turning a coin over is modelled by applying the Pauli-X gate $X = |0\rangle\langle 1| + |1\rangle\langle 0|$. A projective measurement $\{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\}$ with $\mathbf{M}_{\mathrm{t}} = |0,1\rangle\langle 0,1|$ and $\mathbf{M}_{\mathrm{nt}} = |0,0\rangle\langle 0,0| + |1,0\rangle\langle 1,0| + |1,1\rangle\langle 1,1|$ is designed to observe whether the event "the left coin is head and the right is tail" or the complement event happens.

Input: $|q_1, q_2\rangle := |1, 1\rangle$;
1: **while** $\mathbf{M}[q_1, q_2] = \mathrm{nt}$ **do**
2: $\quad (H \otimes X)[q_1, q_2]; \quad \square \quad (X \otimes H)[q_1, q_2]$;

The symbol $\square$ denotes a nondeterministic choice between two coins to be tossed. Once the measurement outcome t occurs under some scheduler, the program terminates. It means that under that scheduler, Alice eventually has the chance of winning, we can infer the protocol is fair.

After setting the entrance of the while loop to be the unique program location, we can formally describe the above program as $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$, where

- $\Sigma = \{\alpha_1, \alpha_2\}$ correspond the choices between the two coins to be tossed;
- $\mathcal{E}(\alpha_1) = \mathcal{E}_1 = \{H \otimes X\}$ and $\mathcal{E}(\alpha_2) = \mathcal{E}_2 = \{X \otimes H\}$.

We would use $\mathcal{F}_{\alpha_1}$ as an abbreviation of $\mathcal{E}_1 \circ \{\mathbf{M}_{\mathrm{nt}}\}$ and $\mathcal{F}_{\alpha_2}$ for $\mathcal{E}_2 \circ \{\mathbf{M}_{\mathrm{nt}}\}$. $\square$

Definition 3 (Termination Probability). *For a nondeterministic quantum program $\mathcal{P}$ defined in Definition 2 and an input state ρ,*

1. *the termination probability along with a finite scheduler* ς *is*

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

2. *the termination probability along with an infinite scheduler* σ *is*

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

3. *the termination probability is* $\mathrm{TP}(\rho) = \inf_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho)$.

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

Based on the notions of program model and termination probability, we would like to consider the following termination problems.

Problem 1 (Universal Termination). Given a nondeterministic quantum program and an input state, does the program terminate with probability one under all schedulers?

Problem 2 (Existential Termination). Given a nondeterministic quantum program and an input state, does the program terminate with probability one under some scheduler?

Problem 3 (Optimal Termination). Given a nondeterministic quantum program and an input state, what is the angelic (resp. demonic) scheduler that maximizes (resp. minimizes) the termination probability?

The first two problems are concerned with qualitative termination, and the last one is on quantitative termination. A program is universally terminating if $\inf_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho) = 1$, while it is existentially terminating if $\sup_{\sigma \in \Sigma^{\omega}} \mathrm{TP}_{\sigma}(\rho) = 1$. We will study Problem 1 in the coming two sections.

4 Computing Reachable Spaces

In this section, we introduce the reachable space for a nondeterministic quantum program starting from an input state, which is crucial in checking whether the program terminates. We first review the notion of reachable space together with the construction method in existing literature [24]. Then we propose a more precise notion of reachable space. The two kinds of reachable spaces are said to be of types I and II respectively, and both are computable in polynomial time.

Definition 4 (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* $\Psi(\mathcal{P}, \rho) = \{\mathcal{F}_{\varsigma}(\rho) : \varsigma \in \Sigma^{*}\}$.

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

Definition 5 (I-Reachable Space, [24, 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 definitions, we can see:

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

Thus, to show that a property holds on the reachable set $\Psi(\mathcal{P},\rho)$, it suffices to show that the property holds on all density operators in $\mathcal{D}(\Phi(\mathcal{P},\rho))$ on the reachable space $\Phi(\mathcal{P},\rho)$. The latter has the algebraic structure of a 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 6 (Average Quantum Program, [24, Definition 4]). *Let $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ 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 $(\bar{\mathcal{E}}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$, 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 1 ([24, 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. $\Phi(\mathcal{P},\rho) = \Phi(\bar{\mathcal{P}},\rho)$.*

Using the above lemma, we have that the I-reachable space of $\mathcal{P}$ can be obtained as the least fixedpoint of the ascending chain of linear subspaces of $\mathbb{H}$:

$$\begin{aligned}\mathrm{supp}(\rho_0) &\subseteq \mathrm{supp}(\rho_0) \vee \mathrm{supp}(\rho_1)\\ &\subseteq \mathrm{supp}(\rho_0) \vee \mathrm{supp}(\rho_1) \vee \mathrm{supp}(\rho_2)\\ &\subseteq \cdots,\end{aligned} \tag{1}$$

where $\rho_i = \bar{\mathcal{F}}^i(\rho_0)$ with $\bar{\mathcal{F}} = \bar{\mathcal{E}} \circ \{\mathbf{M}_{\mathrm{nt}}\}$. Namely, we denote this chain by $\mathbb{B}_0 \subseteq \mathbb{B}_1 \subseteq \mathbb{B}_2 \subseteq \cdots$, in which each linear space $\mathbb{B}_i$ 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 establishes the computability.

Lemma 2. *Let $\mathbb{B}_0 \subseteq \mathbb{B}_1 \subseteq \mathbb{B}_2 \subseteq \cdots$ be the ascending chain of nonnull linear subspaces $\mathbb{B}_i \subseteq \mathbb{H}$, as defined in* (1)*. Then there is an index $\ell \leq \dim(\mathbb{H}) - 2$ such that $\mathbb{B}_k = \mathbb{B}_\ell$ holds for all $k > \ell$.*

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

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

Meanwhile, all subspaces $\mathbb{B}$ of $\mathbb{H}$ form a complete lattice $(\mathbb{B}, \subseteq, \inf, \sup)$ by taking 'inf' as the meet $\bigwedge = \bigcap$ and 'sup' as the joint $\bigvee$. By Knaster–Tarski fixedpoint theorem [8,29], we have that the least fixedpoint occurs upon $\mathbb{B}_\ell = \mathbb{B}_{\ell+1}$, which ℓ is bounded by $\dim(\mathbb{H}) - 2$ since $\mathbb{B}_i$ are nonnull subspaces of $\mathbb{H}$. □

The procedure of computing the I-reachable space $\Phi(\mathcal{P}, \rho_0)$ is stated in Algorithm 1, whose complexity analysis is provided below.

Algorithm 1 Computing I-Reachable Space [24, Algorithm 1]

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_\mathrm{t}, \mathbf{M}_\mathrm{nt}\})$ 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 $\rho_0 \in \mathcal{D}$;
Output: an orthonormal basis B of $\Phi(\mathcal{P}, \rho_0)$.

```
1:  let F̄ = (1/m) Σ_{j=1}^{m} E_j ∘ {M_nt} be the average super-operator;
2:  let {F_j : j = 1, 2, ..., l} be a Kraus representation of F̄;
3:  compute an orthonormal basis B_0 of supp(ρ_0), and B_{-1} ← ∅;
4:  for i ← 0 to d − 2 do
5:      B_{i+1} ← B_i;
6:      for all |ψ⟩ ∈ B_i \ B_{i−1} do
7:          V ← {F_j |ψ⟩ : j = 1, 2, ..., l};
8:          compute an orthonormal basis B′ of V extending to B_{i+1};
9:          B_{i+1} ← B_{i+1} ∪ B′;
10:     if B_{i+1} = B_i or |B_{i+1}| = d then Break;
11: return B_{i+1}.
```

Complexity Note that there are less than $d = \dim(\mathbb{H})$ times of entering the inner loop in Line 6. Each inner loop performs l times of matrix-vector multiplication and l times of computing orthonormal complement, 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 $\mathcal{E}_j$. 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 averaged Kraus operators of $\mathcal{E}_j$, since the simplest Kraus representation is obtained by quantum process tomography [30, Subsection 8.4.2] that costs additionally $\mathcal{O}(d^{12})$ operations. The matrix-vector multiplication is in $\mathcal{O}(d^2)$, and computing orthonormal complement of $\mathbf{F}_j |\psi\rangle$ is also in $\mathcal{O}(d^2)$. Hence Algorithm 1 is in $\mathcal{O}(m \cdot d^5)$. □

Example 2. Consider the nondeterministic quantum program $\mathcal{P}$ in Example 1, the average super-operator is $\bar{\mathcal{F}} = \frac{1}{2}(\mathcal{F}_{\alpha_1} + \mathcal{F}_{\alpha_2}) = \{\mathbf{F}_1, \mathbf{F}_2\}$, in which the Kraus operators are

$$\begin{aligned}
\mathbf{F}_1 &= \tfrac{1}{\sqrt{2}}\mathbf{E}_1\mathbf{M}_\mathrm{nt} = \tfrac{1}{\sqrt{2}}(|+,1\rangle\langle 0,0| + |-,1\rangle\langle 1,0| + |-,0\rangle\langle 1,1|), \\
\mathbf{F}_2 &= \tfrac{1}{\sqrt{2}}\mathbf{E}_1\mathbf{M}_\mathrm{nt} = \tfrac{1}{\sqrt{2}}(|1,+\rangle\langle 0,0| + |0,+\rangle\langle 1,0| + |0,-\rangle\langle 1,1|).
\end{aligned}$$

By Algorithm 1, for the given initial 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 $\mathbb{B}_0 = \mathrm{supp}(\rho_0) = \mathrm{span}(\{|1,1\rangle\})$.
2. To get the next subspace $\mathbb{B}_1$ along the ascending chain, for the basis element $|1,1\rangle$ of $\mathbb{B}_0$, we compute
$$\mathbf{F}_1 |1,1\rangle = \tfrac{1}{\sqrt{2}} |-,0\rangle\,, \qquad \mathbf{F}_2 |1,1\rangle = \tfrac{1}{\sqrt{2}} |0,-\rangle\,.$$
Thus an orthonormal basis extending $\mathbb{B}_0$ is $\{|-,0\rangle\,, (|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}\}$, and $\mathbb{B}_1 = \mathrm{span}(\{|1,1\rangle\,, |-,0\rangle\,, (|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}\})$.
3. To get the next subspace $\mathbb{B}_2$ along the ascending chain, for the newly-produced basis elements $|-,0\rangle$ and $(|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}$ of $\mathbb{B}_1$, we have
$$\begin{aligned}\mathbf{F}_1 |-,0\rangle &= \tfrac{1}{\sqrt{2}} |1,1\rangle\,,\\ \mathbf{F}_2 |-,0\rangle &= -\tfrac{1}{2} |-,+\rangle\,,\\ \mathbf{F}_1(|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3} &= \tfrac{1}{\sqrt{6}} |0,1\rangle\,,\\ \mathbf{F}_2(|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3} &= \tfrac{1}{\sqrt{6}} |+,+\rangle\,.\end{aligned}$$
Thus an orthonormal basis extending $\mathbb{B}_1$ is $\{(-\sqrt{2}\,|+,0\rangle - |0,1\rangle)/\sqrt{3}\}$, and $\mathbb{B}_2 = \mathrm{span}(\{|1,1\rangle\,, |-,0\rangle\,, (|+,0\rangle - \sqrt{2}\,|0,1\rangle)/\sqrt{3}, (-\sqrt{2}\,|+,0\rangle - |0,1\rangle)/\sqrt{3}\})$. Since $\dim(\mathbb{B}_2) = 4 = d = \dim(\mathbb{H})$, we have $\mathbb{B}_2 = \mathbb{H}$.

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

In the following, we will have a deeper study on 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 large superset of the former. So 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 using the linearly independent basis of Hermitian operators on $\mathbb{H}$, say

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

Although the general state is expressed by all d^2 basis elements in (2), all reachable states might be expressed by 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 $\mathcal{H}(\mathbb{H})$ we require. To this end, we resort to the following operator-level program that characterizes the operations between pure reachable states.

Definition 7 (Operator-level Program). *Let $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ be a nondeterministic quantum program. Then the operator-level program $\hat{\mathcal{P}}$ of $\mathcal{P}$ is the triple $(\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$, where*

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

For convenience, we employ the notation $\mathbf{F}_\varsigma$ adapted to $\mathcal{F}_\varsigma$, e.g. $\mathbf{F}_{\alpha_{j,k}} = \mathbf{E}_{j,k}\mathbf{M}_{\mathrm{nt}}$ and $\mathbf{F}_\varsigma = \mathbf{F}_{\varsigma\downarrow 1}\mathbf{F}_{\varsigma\uparrow 1}$.

Definition 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* $\tilde{\Phi}(\mathcal{P}, \rho) = \mathrm{span}(\Psi(\hat{\mathcal{P}}, \rho))$*, where* $\hat{\mathcal{P}}$ *is the operator-level program of* $\mathcal{P}$ *as in Definition 7.*

It is not hard to see that the reachable set $\Psi(\mathcal{P}, \rho)$ is over-approximated by the II-reachable space $\tilde{\Phi}(\mathcal{P}, \rho)$, 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) $\tilde{\Phi}(\mathcal{P}, \rho) = \mathrm{span}(\Psi(\hat{\mathcal{P}}, \rho))$.

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 $\mathcal{H}(\mathbb{H})$:

$$\begin{aligned} \mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| = 0\}) &\subseteq \mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| \leq 1\}) \\ &\subseteq \mathrm{span}(\{\{\mathbf{F}_\varsigma\}(\rho) : \varsigma \in \hat{\Sigma}^* \wedge |\varsigma| \leq 2\}) \\ &\subseteq \cdots . \end{aligned} \tag{3}$$

The following lemma gives an upper bound for the occurrence of the least fixedpoint in the ascending chain.

Lemma 3. *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* (3)*. Then there is an index* $\ell \leq \dim(\mathbb{H})^2 - 2$ *such that* $\Theta_k = \Theta_\ell$ *holds for all* $k > \ell$*.*

Proof. The proof is similar to that of Lemma 2. The function G from Θ_i to Θ_{i+1} $(i \geq 0)$ can be formulated as a monotonic function

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

Meanwhile, all subspaces $\mathbf{\Theta}$ of $\mathcal{H}(\mathbb{H})$ form a complete lattice $(\mathbf{\Theta}, \subseteq, \inf, \sup)$ by taking 'inf' as the meet $\bigwedge = \bigcap$ and 'sup' as the joint $\bigvee$. By Knaster–Tarski fixedpoint theorem [8,29], we have that the least fixedpoint occurs upon $\Theta_\ell = \Theta_{\ell+1}$, where ℓ is bounded by $\dim(\mathbb{H})^2 - 2$ since Θ_i are nonnull subspaces of $\mathcal{H}(\mathbb{H})$. $\square$

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

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

Algorithm 2 Computing II-Reachable Space

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \dots, m\}$, $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ and $\mathcal{E}_j = \{\mathbf{E}_{j,k} : k = 1, 2, \dots, K_j\}$ 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 $\tilde{\Phi}(\mathcal{P}, \rho_0)$ whose elements are pure states.

let $\hat{\Sigma} = \{\alpha_{j,k} : j = 1, 2, \dots, m \wedge k = 1, 2, \dots, K_j\}$, and $\mathbf{E}(\alpha_{j,k}) = \mathbf{E}_{j,k}$;

let $\hat{\mathcal{P}} = (\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ be the operator-level program of $\mathcal{P}$;

$\mathbf{F}_{\alpha_{j,k}} \leftarrow \mathbf{E}_{j,k}\mathbf{M}_{\mathrm{nt}}$ with $j = 1, 2, \dots, m$ and $k = 1, 2, \dots, K_j$;

$B_0 \leftarrow \{|\lambda\rangle\}$, $B_{-1} \leftarrow \emptyset$, and $\theta_0 \leftarrow \{\rho_0\}$;

for $i \leftarrow 0$ to $d^2 - 2$ **do**

  $B_{i+1} \leftarrow B_i$ and $\theta_{i+1} \leftarrow \theta_i$;

  for all $|\psi\rangle \in B_i \setminus B_{i-1}$ **do**

    $V \leftarrow \{\mathbf{F}_{\alpha_{j,k}} |\psi\rangle / \|\mathbf{F}_{\alpha_{j,k}} |\psi\rangle\| : j = 1, 2, \dots, m \wedge k = 1, 2, \dots, K_j\}$;

    find a maximal subset B' of V, such that $\theta' = \{|\psi'\rangle\langle\psi'| : |\psi'\rangle \in B'\}$ is a linearly independent basis extending to θ_{i+1};

    $B_{i+1} \leftarrow B_{i+1} \cup B'$ and $\theta_{i+1} \leftarrow \theta_{i+1} \cup \theta'$;

  if $B_{i+1} = B_i$ or $|B_{i+1}| = d^2$ **then Break**;

return θ_{i+1}.

multiplication is in $\mathcal{O}(d^2)$, the normalization is in $\mathcal{O}(d)$, and checking the linear independence can be in $\mathcal{O}(d^4)$ by embedding with the orthonormalization of the linearly independent basis, i.e. the output linearly independent basis θ induces an orthonormal basis, in which each element can be obtained in $\mathcal{O}(d^4)$ by the Gram–Schmit procedure. Hence Algorithm 2 is in $\mathcal{O}(m \cdot d^8)$. □

Example 3. Reconsider the program $\mathcal{P}$ in Example 2, the operator-level program $\hat{\mathcal{P}} = (\hat{\Sigma}, \mathbf{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ of $\mathcal{P}$ provides

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

We define $\mathbf{F}_{\alpha_{1,1}} = \mathbf{E}_{1,1}\mathbf{M}_{\mathrm{nt}}$ and $\mathbf{F}_{\alpha_{2,1}} = \mathbf{E}_{2,1}\mathbf{M}_{\mathrm{nt}}$. 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

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

   So we have $V = \{|-,0\rangle, |0,-\rangle\}$. Since the two pure states in V have density operators that form a linearly independent basis extending 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

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

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

It is not hard to see that $\Phi(\mathcal{P}, \rho_0)$ contains all pure states in $\mathbb{H}$ while $\tilde{\Phi}(\mathcal{P}, \rho_0)$ has dimension 8 that is less than $\dim(\mathcal{H}(\mathbb{H})) = 16$. Hence there are many pure states in $\Phi(\mathcal{P}, \rho_0)$ whose density operators are not in $\tilde{\Phi}(\mathcal{P}, \rho_0)$, e.g. the pure state $|\varphi\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$ in $\Phi(\mathcal{P}, \rho_0)$ cannot be linearly expressed by the basis of $\tilde{\Phi}(\mathcal{P}, \rho_0)$. The II-reachable space $\tilde{\Phi}(\mathcal{P}, \rho_0)$ gives an over-approximation of $\Psi(\mathcal{P}, \rho_0)$ more precise than $\Phi(\mathcal{P}, \rho_0)$ in this example. □

Remark 1. The ascending chain $\Theta_0 \subseteq \Theta_1 \subseteq \Theta_2 \subseteq \cdots$ as in (3) is finer than the ascending chain $\mathbb{B}_0 \subseteq \mathbb{B}_1 \subseteq \mathbb{B}_2 \subseteq \cdots$ as in (1) in such a sense:

- For each linear subspace $\Theta_i \subseteq \mathcal{H}(\mathbb{H})$, there is a unique index j such that $\Theta_i \subseteq \mathcal{H}(\mathbb{B}_j)$ and $\Theta_i \not\subseteq \mathcal{H}(\mathbb{B}_{j-1})$.
- For each linear subspace $\mathbb{B}_j \subseteq \mathbb{H}$, there are some indices i such that $\Theta_i \subseteq \mathcal{H}(\mathbb{B}_j)$ and $\Theta_i \not\subseteq \mathcal{H}(\mathbb{B}_{j-1})$.
- 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 $\mathcal{D}(\mathbb{B}_j)$.

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

By Algorithms 1 and 2, we obtain the result:

Theorem 1. *Both I-reachable space and II-reachable space are computable in polynomial time.*

5 Computing Diverging Set

In this section, we compute the set of *divergent* states from which a given nondeterministic quantum program terminates with probability zero under some scheduler. The procedure turns out to be in exponential time. Combining the divergent set with the reachable spaces, we are able to analyze the universal termination of the nondeterministic quantum program.

Definition 9. *Given a nondeterministic quantum program $\mathcal{P}$ with the quantum state space $\mathbb{H}$,*

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

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

The divergence requires that all eigenstates in $\mathrm{supp}(\rho)$ terminate with probability zero. It is not hard to see that an element in the divergent set D is an ensemble of some elements in the divergent set PD, and vice verse. Once the pure divergent set PD is determined, the divergent set D is also determined. So we would only focus on PD.

For convenience, we would like to introduce some auxiliary notations:

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

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

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

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

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

From the above definitions and notions, we can see:

- for any infinite scheduler σ and any integer i, PD_i^σ is a subspace of $\mathbb{H}$ [24, Lemma 4], and $PD_i^\sigma \supseteq PD_{i+1}^\sigma$, as the latter requires that the program does not terminate at one more step;
- 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, $PD_i = \bigcup_{\sigma\in\Sigma^\omega} PD_i^\sigma$ is a finite union of subspaces, as there are only finitely many distinct i-fragments ς of all infinite schedulers σ; and
- $PD = \bigcap_{i=0}^\infty PD_i = \lim_{i\to\infty} PD_i$.

Particularly, we have $PD_0 = PD^\epsilon = \{|\psi\rangle \in \mathbb{H} : \mathbf{M}_{\mathrm{t}}|\psi\rangle = 0\}$; and for a subspace $PD^\varsigma \subseteq PD_i$ and an action $\alpha \in \Sigma$, we can calculate:

$$\begin{aligned} PD^{\alpha\cdot\varsigma} &= \{|\psi\rangle \in PD_0 : \mathcal{F}_\alpha(|\psi\rangle\langle\psi|) \in \mathcal{D}(PD^\varsigma)\} \\ &= \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PD^\varsigma\}, \end{aligned} \tag{4}$$

where $\alpha\cdot\varsigma$ denotes the concatenation of α and ς that takes ς as a suffix, not a prefix. We collect all subspaces $PD^{\alpha\cdot\varsigma}$ with α ranging over Σ and ς ranging over Σ^i as PD_{i+1}, i.e.

$$\begin{aligned} PD_{i+1} &= \bigcup_{\alpha\in\Sigma}\bigcup_{\varsigma\in\Sigma^i} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PD^\varsigma\} \\ &= \bigcup_{\varsigma\in\Sigma^i} \{|\psi\rangle \in PD_0 : \mathrm{supp}(\mathcal{F}_\alpha(|\psi\rangle\langle\psi|)) \subseteq PD^\varsigma \wedge \alpha \in \Sigma\}. \end{aligned} \tag{5}$$

Note that the set PD_{i+1} depends on the prior set PD_i.

We notice that the derivation of those sets PD_i can be organized as an infinite m-branching tree (see Fig. 1), in which

- the root is labelled with the empty scheduler ϵ representing the subspace $PD^\epsilon = PD_0$; and
- each intermediate node with label ς representing the subspace PD^ς has m children with labels $\varsigma\cdot\alpha$ ($\alpha \in \Sigma$) representing the subspaces $PD^{\varsigma\cdot\alpha}$.

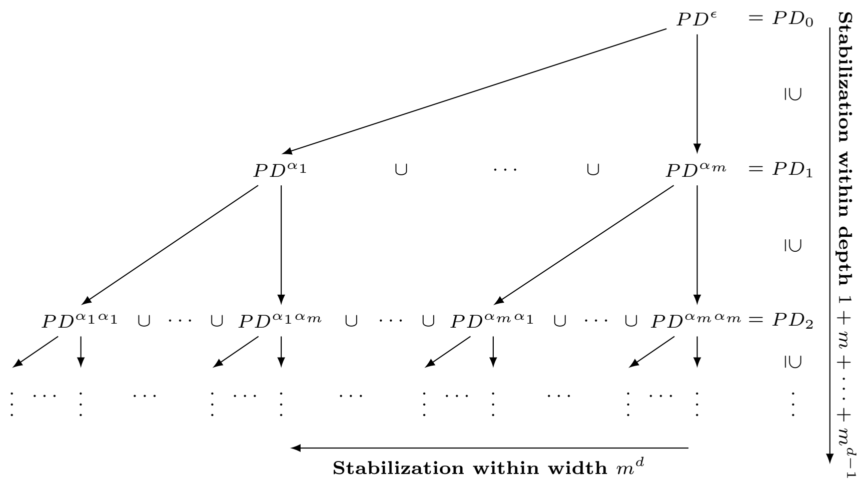

Fig. 1. Derivation of PD_i by a tree construction

Thus, the union of the subspaces generated 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 the subspace PD^{ς} generated by an intermediate node is a common superset of the subspaces $PD^{\varsigma\cdot\alpha}$ with $\alpha \in \Sigma$ generated by the m children of that intermediate node.

The following lemma gives an upper bound for the occurrence of the least fixedpoint in the descending chain of finite unions of subspaces of $\mathbb{H}$.

Lemma 4. *Let $PD_0 \supseteq PD_1 \supseteq PD_2 \supseteq \cdots$ be a descending chain of finite unions of nonempty subspaces $PD_i \subseteq \mathbb{H}$, as defined in* (4)*. Then there is an index $\ell < M = 1 + m + \cdots + m^{d-1}$ such that $PD_k = PD_\ell$ holds for all $k > \ell$.*

Proof. The proof is an extension to that of [24, Lemma 6] by giving the explicit bound M. We first prove the existence of such a least fixedpoint PD_ℓ by an induction on the dimension of PD_0.

- Basically, when $\dim(PD_0) = 0$, we have $PD_0 = \{0\}$. It is plainly the fixedpoint of the chain, as the pure divergent set PD is empty then.
- Inductively, when $\dim(PD_0) > 0$, we, again, assume that PD_0 is not the fixedpoint of the chain; as otherwise it is trivial. Then there is a least index l such that $PD_l \neq PD_0$. Let $PD_l = \bigcup_{i=1}^m P_i$ where P_i are subspaces. Define $Z_{k,i} = PD_k \cap P_i$ for $k \geq l$. We have $PD_k = \bigcup_{i=1}^m Z_{k,i}$ with $k \geq l$ and the following m descending chains:

$$P_1 = Z_{l,1} \supseteq Z_{l+1,1} \supseteq Z_{l+2,1} \supseteq \cdots$$
$$\cdots$$
$$P_m = Z_{l,m} \supseteq Z_{l+1,m} \supseteq Z_{l+2,m} \supseteq \cdots .$$

As PD_0 is a single subspace, we have $\dim(P_i) < \dim(PD_0)$. By induction hypothesis, we know there is a fixedpoint $Z_{\ell_i,i}$ in the above ith chain. Finally, letting $\ell = \max_{i=1}^{m} \ell_i$, PD_ℓ is the fixedpoint of the original chain, since $PD_\ell = \bigcup_{i=1}^{m} Z_{\ell,i} = \bigcup_{i=1}^{m} Z_{k,i} = PD_k$ holds for all $k > \ell$.

Then, we can see that the least fixedpoint occurs upon $PD_{\ell+1} = PD_\ell$, since

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

and $PD_k = PD_\ell$ follows for all $k > \ell + 2$ similarly. We further show that the index ℓ of the least fixedpoint PD_ℓ can be bounded by $M - 1$. It follows from the derivation tree that there are at most M strictly descending layers from $PD_0 \subseteq \mathbb{H}$ (the full space) to $PD_M \supseteq \{0\}$ (the null space). $\square$

The above lemma also indicates that the derivation tree is stabilized with height bounded by M and width bounded by m^d by removing those intermediate nodes whose representing subspaces are contained by those of their brothers.

The procedure of computing the pure divergent set PD is stated in Algorithm 3, whose complexity analysis is provided below.

Algorithm 3 Computing Pure Diverging Set

Input: a nondeterministic quantum program $\mathcal{P} = (\Sigma, \mathcal{E}, \{\mathbf{M}_{\mathrm{t}}, \mathbf{M}_{\mathrm{nt}}\})$ with $\Sigma = \{\alpha_j : j = 1, 2, \dots, m\}$ and $\mathcal{E}(\alpha_j) = \mathcal{E}_j$ over $\mathbb{H}$ with dimension d;
Output: a set Z of finite schedulers that generates the pure divergent set PD of $\mathcal{P}$.

```
let F_{α_j} = E_j ∘ {M_nt} with j = 1,...,m be the composite super-operators;
compute the subspace PD_0 = {|ψ⟩ ∈ H : M_t |ψ⟩ = 0};
Z_0 ← {ε};
for i ← 0 to M − 2 do
    Z_{i+1} ← ∅;
    for j ← 1 to m do
        Z' ← Z_i;
        while Z' ≠ ∅ do
            let ς be an element of Z', and ς' ← (α_j · ς) ↑ i;
            compute the subspace PD^{α_j·ς} = {|ψ⟩ ∈ PD_0 : supp(F_{α_j}(|ψ⟩⟨ψ|)) ⊆ PD^ς};
            Z_{i+1} ← Z_{i+1} ∪ {α_j · ς};
            if PD^{α_j·ς} = PD^{ς'} then
                remove all elements with prefix ς ↑ (i − 1) from Z';
            else Z' ← Z' \ {ς};
    PD_{i+1} ← ⋃_{ς∈Z_{i+1}} PD^ς;
    if PD_{i+1} = PD_i or PD_{i+1} = {0} then Break;
return Z_{i+1}.
```

Complexity Note that there are less than $M = 1+m+\cdots+m^{d-1}$ times of entering the inner loop in Line 8. Each inner loop needs to compute the subspace $PD^{\alpha_j \cdot \varsigma}$ in Line 10. It can be obtained in such a way: we first introduce at most $2d$ real variables to encode $|\psi\rangle$ as a parametric linear combination of basis elements of PD_0; then the predicate $\mathrm{supp}(\mathcal{F}_{\alpha_j}(|\psi\rangle\langle\psi|)) \subseteq PD^{\varsigma}$ results in a polynomial formula with those real variables; finally we solve the polynomial formula in $2^{\mathcal{O}(d)}$ by the existential theory of the reals [5, Theorem 13.13] that is in exponential time w.r.t. the number of real variables. Hence Algorithm 3 is in exponential time $2^{\mathcal{O}(d)}$ due to $M \in 2^{\mathcal{O}(d)}$. The exponential hierarchy seems to be tight, since there are two bottlenecks that are in exponential time. □

Example 4. We compute the pure divergent set PD of program $\mathcal{P}$ in Example 1. The pure divergent set can be inductively computed as follows.

1. Initially, we have $PD_0 = PD^{\epsilon} = \mathrm{span}(\{|0,0\rangle, |1,0\rangle, |1,1\rangle\})$.
2. For actions α_1 and α_2, we compute
$$\begin{aligned} PD^{\alpha_1} &= \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}), \\ PD^{\alpha_2} &= \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}). \end{aligned}$$
Thus, we get
$$PD_1 = PD^{\alpha_1} \cup PD^{\alpha_2} = \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}) \cup \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}).$$
3. Next, we compute
$$\begin{aligned} PD^{\alpha_1\alpha_1} &= \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}), \\ PD^{\alpha_2\alpha_1} &= \mathrm{span}(\{(-\sqrt{2}\,|1,1\rangle + |-,0\rangle)/\sqrt{3}\}), \\ PD^{\alpha_1\alpha_2} &= \mathrm{span}(\{(-\,|0,0\rangle + \sqrt{2}\,|1,+\rangle)/\sqrt{3}\}), \\ PD^{\alpha_2\alpha_2} &= \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}). \end{aligned}$$
Thus, we get
$$\begin{aligned} PD_2 &= PD^{\alpha_1\alpha_1} \cup PD^{\alpha_2\alpha_1} \cup PD^{\alpha_1\alpha_2} \cup PD^{\alpha_2\alpha_2} \\ &= \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}) \cup \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}) = PD_1. \end{aligned}$$

Hence, the least fixedpoint of the descending chain occurs, which yields the pure divergent set $PD = PD_2$. □

By Algorithm 3, we obtain the result:

Theorem 2. *Both pure divergent set and divergent set are computable in exponential time.*

Finally, we combine the results on reachability and divergence to analyze the universal termination of a nondeterministic quantum program $\mathcal{P}$ with an input state ρ. To refute the universal termination, a necessary and sufficient condition is finding an infinite scheduler σ under which the termination probability is less than 1, i.e. $\lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\mathcal{F}_{\sigma\uparrow i}(\rho)) > 0$. The following lemma indicates that the pure divergent set is a small-model of this condition. The small-model property means the former set is nonempty if and only if the latter is nonempty.

Lemma 5. *Given a nondeterministic quantum program $\mathcal{P}$ and an input state $\rho \in \mathcal{D}$, $\mathcal{P}$ is not universally terminating on ρ if and only if there is a pure divergent state $|\psi\rangle$ falling into the support of a reachable state γ from ρ under some infinite scheduler σ.*

Proof. We first prove the "if" direction by the following construction. Let ς be a finite scheduler such that $\gamma = \mathcal{F}_\varsigma(\rho)$, and $|\psi\rangle$ an element of $\mathrm{supp}(\gamma)$. Then, by [30, Exercise 2.73], there is an ensemble of γ containing $|\psi\rangle$ with positive probability p. By the definition of PD, there is an infinite scheduler σ' such that $\lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\mathcal{F}_{\sigma'\uparrow i}(|\psi\rangle\langle\psi|)) = 1$. So, letting $\sigma = \varsigma \cdot \sigma'$, we have

$$\begin{aligned}\lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\mathcal{F}_{(\varsigma\cdot\sigma')\uparrow i}(\rho)) &= \lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\mathcal{F}_{\sigma'\uparrow i}(\gamma)) \\ &\geq \lim_{i\to\infty} \mathrm{tr}(\mathbf{M}_{\mathrm{nt}}\mathcal{F}_{\sigma'\uparrow i}(p\,|\psi\rangle\langle\psi|)) = p,\end{aligned}$$

which entails that $\mathcal{P}$ does not terminate with probability 1 on ρ under the infinite scheduler σ, i.e. it is not universally terminating on ρ.

For the "only if" direction, we assume that $\mathcal{P}$ is not universally terminating on ρ. Then, there is an infinite scheduler σ, such that from ρ the program has a positive probability of nontermination. This condition implies:

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

Since those eigenstates $|\lambda_0\rangle, |\lambda_1\rangle, |\lambda_2\rangle, \ldots$ are unit vectors falling into the supports of some reachable states, there is a convergent subsequence of $|\lambda_0\rangle, |\lambda_1\rangle, |\lambda_2\rangle, \ldots$ falling into the support of a fixed reachable state. By the completeness of Hilbert space that the limit of a convergent sequence is contained in that space, the limit $|\lambda\rangle$ of the subsequence is in $\mathbb{H}$, which falls into the support of some reachable state and is a pure divergent state as $|\lambda\rangle$ has nontermination probability $\lim_{i\to\infty} p_i = 1$. $\square$

Using the above lemma, we can safely conclude that a nondeterministic quantum program is universally terminating if the reachable space and the divergent set are disjoint in terms of pure states $\mathbb{H}$ or ensembles $\mathcal{D}(\mathbb{H})$.

- To check the emptiness of $\Psi(\mathcal{P},\rho) \cap PD(\mathcal{P})$, we compute the intersection of $\Psi(\mathcal{P},\rho)$ and PD^{ς} for each $PD^{\varsigma} \in PD(\mathcal{P})$. It can be solved in exponential time as there are at most m^{d-1} subspaces PD^{ς} in $PD(\mathcal{P})$.
- To check the emptiness of $\tilde{\Psi}(\mathcal{P},\rho) \cap \mathcal{D}(PD(\mathcal{P}))$, we try to find a pure state $|\psi\rangle \in PD^{\varsigma}$ that falls into the support of some element in $\tilde{\Psi}(\mathcal{P},\rho)$ for each $PD^{\varsigma} \in PD(\mathcal{P})$. It is also solved in exponential time as there are at most m^{d-1} subspaces PD^{ς} in $PD(\mathcal{P})$ and these $|\psi\rangle$ can be obtained in exponential time $2^{\mathcal{O}(d^2)}$ by the existential theory of the reals [5, Theorem 13.13].

Example 5. For the program $\mathcal{P}$ and the initial state ρ_0 in Example 1, we have obtained the I/II-reachable spaces and the pure divergent set in the previous examples. Then we compute the intersections as follows.

$$\begin{aligned} \Psi(\mathcal{P},\rho_0) \cap PD(\mathcal{P}) &= \mathrm{span}(\{|1,1\rangle, |-,0\rangle\}) \cup \mathrm{span}(\{|0,0\rangle, |1,+\rangle\}), \\ \tilde{\Psi}(\mathcal{P},\rho_0) \cap \mathcal{D}(PD(\mathcal{P})) &= \mathcal{D}(\{|1,1\rangle\}) \cup \mathcal{D}(\{|-,0\rangle\}) \cup \mathcal{D}(\{|0,0\rangle, |1,+\rangle\}). \end{aligned}$$

Both are not null, thus we cannot infer the universal termination. However, it can be seen that the input state $|1,1\rangle\langle 1,1|$ is a pure divergent one as $|1,1\rangle\langle 1,1| \in \mathcal{D}(PD(\mathcal{P}))$. Therefore the program $\mathcal{P}$ is not universally terminating, i.e., the protocol is proved to be unfair.

6 Conclusion

In this paper, we have studied the model of nondeterministic quantum program and its universal termination problem. We achieved this goal by two parts. One was computing the reachable space of a program with an input state, that is a superset of the set of reachable states but was of explicit algebraic structure. A more precise characterization of reachable space 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. Once the two sets were disjoint, we could safely infer the universal termination. A case study of the quantum Bernoulli factory protocol was provided to demonstrate our method.

For future work, we would like to:

- explore more precise characterization of reachable space using explicit algebraic structure toward the completeness,
- design more efficient algorithms for computing the divergent set, and
- consider the existential termination and the optimal termination over nondeterministic quantum programs, as listed in Problems 2 & 3.

References

1. Agrawal, S., Chatterjee, K., Novotný, P.: Lexicographic ranking supermartingales: An efficient approach to termination of probabilistic programs. Proceedings of the ACM on Programming Languages **2**(POPL), 34:1–34:32 (2018)
2. Altenkirch, T., Grattage, J.: A functional quantum programming language. In: Proc. 20th IEEE Symposium on Logic in Computer Science (LICS 2005). pp. 249–258. IEEE Computer Society (2005)
3. Ardeshir-Larijani, E., Gay, S.J., Nagarajan, R.: Verification of concurrent quantum protocols by equivalence checking. In: Ábrahám, E., Havelund, K. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014. LNCS, vol. 8413, pp. 500–514. Springer (2014)
4. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M.P., Hartmann, M.J., Ho, A., Hoffmann, M., Huang, T., Humble, T.S., Isakov, S.V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P.V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrá, S., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M.Y., Ostby, E., Petukhov, A., Platt, J.C., Quintana, C., Rieffel, E.G., Roushan, P., Rubin, N.C., Sank, D., Satzinger, K.J., Smelyanskiy, V., Sung, K.J., Trevithick, M.D., Vainsencher, A., Villalonga, B., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Neven, H., Martinis, J.M.: Quantum supremacy using a programmable superconducting processor. Nature **574**, 505–510 (2019)
5. Basu, S., Pollack, R., Roy, M.F.: Algorithms in Real Algebraic Geometry. Springer, 2 edn. (2006)
6. Chakarov, A., Sankaranarayanan, S.: Probabilistic program analysis with martingales. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification, 25th International Conference, CAV 2013. LNCS, vol. 8044, pp. 511–526. Springer (2013)
7. Chatterjee, K., Fu, H., Goharshady, A.K.: Termination analysis of probabilistic programs through positivstellensatz's. In: Chaudhuri, S., Farzan, A. (eds.) Computer Aided Verification, 28th International Conference, CAV 2016, Part I. LNCS, vol. 9779, pp. 3–22. Springer (2016)
8. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Graham, R.M., Harrison, M.A., Sethi, R. (eds.) Proc. 4th ACM Symposium on Principles of Programming Languages. pp. 238–252. ACM (1977)
9. Feng, Y., Yu, N., Ying, M.: Model checking quantum Markov chains. Journal of Computer and System Sciences **79**(7), 1181–1198 (2013)
10. Feng, Y., Hahn, E.M., Turrini, A., Zhang, L.: QPMC: A model checker for quantum programs and protocols. In: Bjørner, N., de Boer, F.S. (eds.) FM 2015: Formal Methods - 20th International Symposium. LNCS, vol. 9109, pp. 265–272. Springer (2015)
11. Fioriti, L.M.F., Hermanns, H.: Probabilistic termination: Soundness, completeness, and compositionality. In: Rajamani, S.K., Walker, D. (eds.) Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015. pp. 489–501. ACM (2015)

12. Fu, H., Chatterjee, K.: Termination of nondeterministic probabilistic programs. In: Enea, C., Piskac, R. (eds.) Verification, Model Checking, and Abstract Interpretation - 20th International Conference, VMCAI 2019. LNCS, vol. 11388, pp. 468–490. Springer (2019)
13. Gay, S.J.: Quantum programming languages: survey and bibliography. Mathematical Structures in Computer Science **16**(4), 581–600 (2006)
14. Gay, S.J., Nagarajan, R., Papanikolaou, N.: QMC: A model checker for quantum systems. In: Gupta, A., Malik, S. (eds.) Computer Aided Verification, 20th International Conference, CAV 2008. LNCS, vol. 5123, pp. 543–547. Springer (2008)
15. Google: Cirq: A Python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators. https://github.com/quantumlib/Cirq (2018)
16. Green, A.S., Lumsdaine, P.L., Ross, N.J., Selinger, P., Valiron, B.: Quipper: a scalable quantum programming language. In: Boehm, H., Flanagan, C. (eds.) ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2013. pp. 333–342. ACM (2013)
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. 28th Annual ACM Symposium on the Theory of Computing. pp. 212–219. ACM (1996)
18. He, J., Seidel, K., McIver, A.: Probabilistic models for the guarded command language. Science of Computer Programming **28**(2–3), 171–192 (1997)
19. Hoare, C.A.R.: An axiomatic basis for computer programming. Communications of the ACM **12**(10), 576–580 (1969)
20. IBM: Qiskit: An open-source SDK for working with quantum computers at the level of pulses, circuits, and algorithms. https://github.com/QISKit (2020)
21. Kaminski, B.L., Katoen, J.P., Matheja, C., Olmedo, F.: Weakest precondition reasoning for expected run-times of probabilistic programs. In: Thiemann, P. (ed.) Programming Languages and Systems, 25th European Symposium on Programming, ESOP 2016. LNCS, vol. 9632, pp. 364–389. Springer (2016)
22. Keane, M.S., O'Brien, G.L.: A Bernoulli factory. ACM Transactions on Modeling and Computer Simulation **4**(2), 213–219 (1994)
23. Li, Y., Ying, M.: Algorithmic analysis of termination problems for quantum programs. Proceedings of the ACM on Programming Languages **2**(POPL), 35:1–35:29 (2018)
24. Li, Y., Yu, N., Ying, M.: Termination of nondeterministic quantum programs. Acta Informatica **51**(1), 1–24 (2014)
25. Liu, J., Zhan, B., Wang, S., Ying, S., Liu, T., Li, Y., Ying, M., Zhan, N.: Formal verification of quantum algorithms using quantum Hoare logic. In: Dillig, I., Tasiran, S. (eds.) Computer Aided Verification - 31st International Conference, CAV 2019, Part II. LNCS, vol. 11562, pp. 187–207. Springer (2019)
26. McIver, A., Morgan, C., Kaminski, B.L., Katoen, J.: A new proof rule for almost-sure termination. Proceedings of the ACM on Programming Languages **2**(POPL), 33:1–33:28 (2018)
27. McIver, A., Morgan, C.: Abstraction, Refinement and Proof for Probabilistic Systems. Springer (2006)
28. Moosbrugger, M., Bartocci, E., Katoen, J., Kovács, L.: Automated termination analysis of polynomial probabilistic programs. In: Yoshida, N. (ed.) Programming Languages and Systems - 30th European Symposium on Programming, ESOP 2021. LNCS, vol. 12648, pp. 491–518. Springer (2021)
29. Müller-Olm, M., Seidl, H.: Computing polynomial program invariants. Information Processing Letters **91**(5), 233–244 (2004)

30. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
31. Ömer, B.: A procedural formalism for quantum computing. Tech. rep., Technical University of Vienna (1998)
32. Qin, X., Deng, Y., Du, W.: Verifying quantum communication protocols with ground bisimulation. In: Biere, A., Parker, D. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference, TACAS 2020, Part II. LNCS, vol. 12079, pp. 21–38. Springer (2020)
33. Rigetti: A Python library for quantum programming using Quil. https://github.com/rigetti/pyquil (2017)
34. Sanders, J.W., Zuliani, P.: Quantum programming. In: Backhouse, R., Oliveira, J.N. (eds.) Mathematics of Program Construction. pp. 80–99. Springer (2000)
35. Selinger, P.: A brief survey of quantum programming languages. In: Kameyama, Y., Stuckey, P.J. (eds.) Functional and Logic Programming, 7th International Symposium, FLOPS 2004. LNCS, vol. 2998, pp. 1–6. Springer (2004)
36. Selinger, P.: Towards a quantum programming language. Mathematical Structures in Computer Science **14**(4), 527–586 (2004)
37. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proc. 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society (1994)
38. Svore, K., Geller, A., Troyer, M., Azariah, J., Granade, C., Heim, B., Kliuchnikov, V., Mykhailova, M., Paz, A., Roetteler, M.: Q#: Enabling scalable quantum computing and development with a high-level DSL. In: Proceedings of the Real World Domain Specific Languages Workshop. pp. 1–10. ACM (2018)
39. Tavala, A.M., Nazem, S., Babaei-Brojeny, A.A.: Verification of quantum protocols with a probabilistic model-checker. Electronic Notes in Theoretical Computer Science **270**(1), 175–182 (2011)
40. Ying, M.: Toward automatic verification of quantum programs. Formal Aspects of Computing **31**(1), 3–25 (2019)
41. Ying, M., Yu, N., Feng, Y., Duan, R.: Verification of quantum programs. Science of Computer Programming **78**(9), 1679–1700 (2013)
42. Ying, M.: Floyd-Hoare logic for quantum programs. ACM Transactions on Programming Languages and Systems **33**(6), 19:1–19:49 (2011)
43. Ying, M., Feng, Y.: Quantum loop programs. Acta Informatica **47**(4), 221–250 (2010)
44. Yu, N., Ying, M.: Reachability and termination analysis of concurrent quantum programs. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012 - Concurrency Theory - 23rd International Conference. LNCS, vol. 7454, pp. 69–83. Springer (2012)
45. Zhong, H., Wang, H., Deng, Y., Chen, M., Peng, L., Luo, Y., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X., Zhang, W., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N., Lu, C., Pan, J.: Quantum computational advantage using photons. Science **370**(6523), 1460–1463 (2020)

A Implementation

The prototypes of Algorithms 1, 2 and 3 have been implemented in the Wolfram language on Mathematica 11.3 with Intel Core i7-10700 CPU at 2.90GHz. The source files are available at https://github.com/Holly-Jiang/TANQPR. All the functions required for analyzing the termination of a nondeterministic quantum program are listed as follows.

- `Initialization.nb` initializes a nondeterministic program with given information about super-operators, projective measurement and an input state.
- `ReachableSpaceI.nb` computes the I-reachable subspace w.r.t. an input state and returns an orthonormal basis of that subspace of Hilbert space.
- `ReachableSpaceII.nb` computes the II-reachable subspace w.r.t. an input state and returns a linearly independent basis of that subspace of Hermitian operators on Hilbert space. In particular, we make use of the function `LinearIndepHerm` that checks whether a Hermitian operator can be linearly expressed by the current linearly independent basis;
- `DivergentSet.nb` computes the set of pure divergent states from which the given nondeterministic quantum program terminates with probability zero under some scheduler.
  - `SpaceUnionNull` checks whether the union of subspaces is null;
  - `SpaceUnionEqual` checks whether two unions of subspaces are equal;
  - `PDSpace` computes the subspace of all pure divergent states under a given scheduler;
  - `ISpaceIntersectEmpty` (resp. `IISpaceIntersectEmpty`) checks whether the I-reachable (resp. II-reachable) subspace is disjoint with the pure divergent set.

After fixing the dimension of the Hilbert space, a nondeterministic quantum programs, and an input state, one can invoke the algorithms by calling the above functions respectively.

Generally speaking, all the functions in the files `ReachableSpaceI.nb` and `ReachableSpaceII.nb` are efficient as their theoretical complexity is **PTIME**. They take time 16ms, 15ms and space 104.40MB, 103.51MB, respectively on the running example. Those in the file `divergentSet.nb` may be inefficient (in the worst case), due to the fact that the quantifier elimination and the derivation of the pure divergent set by a tree construction are both **EXPTIME**. However, it fortunately takes time 2797ms and space 105.91MB on our running example.