QReach: A Reachability Analysis Tool for Quantum Markov Chains

Aochu Dai[1] and Mingsheng Ying[1,2(✉)]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
dac22@mails.tsinghua.edu.cn

[2] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
yingms@ios.ac.cn

Abstract. We present QReach, the first reachability analysis tool for quantum Markov chains based on decision diagrams CFLOBDD (presented at *CAV* 2023). QReach provides a novel framework for finding reachable subspaces, as well as a series of model-checking subprocedures like image computation. Experiments indicate its practicality in verification of quantum circuits and algorithms. QReach is expected to play a central role in future quantum model checkers.

Keywords: Reachability analysis · Quantum Markov chain · Quantum Model Checking

1 Introduction

A rapid growth of quantum computing hardware has been witnessed in the last few years. As a recent breakthrough, IBM has introduced its new quantum processor Condor, which breaks the 1000-qubit barrier. Many researchers share the belief that quantum computation will be scalable and stable enough for some meaningful quantum algorithms in the foreseeable future. In the era of Fault-Tolerant Quantum Computing (FTQC), quantum systems may be too complicated to be designed and verified manually. On the other hand, systematic ventures occurring in a quantum circuit or a communication protocol may differ significantly from those in classical systems and may be counterintuitive. The success of model checking techniques in classical computing and communication industry motivates us to extend it for analysis and verification of various temporal properties of a quantum system. Indeed, several model checking algorithms have been proposed for quantum automata and quantum Markov chains [10,11,18]. Additionally, some basic communication protocols like BB84 have passed the verification of the proposed quantum model checkers [2]. However, these quantum model checkers cannot be applied to larger quantum systems.

As is well known, the scalability of classical model checkers heavily relies on the data structures (in particular, various DDs (Decision Diagrams), e.g.

A. Gurfinkel and V. Ganesh (Eds.): CAV 2024, LNCS 14683, pp. 520–532, 2024.
https://doi.org/10.1007/978-3-031-65633-0_23

ROBDD) employed in them for representing the system under checking. Several quantum generalisations of DDs have been introduced for modelling, simulation, and verification of (combinational) quantum circuits, like QMDD [12], TDD [9], and LimDD [17], providing different degrees of compression for quantum states and operators. Based on these diagram structures, some simulation or verification tools were developed for experimental tasks like equivalence checking and bug finding [5,6]. Decision diagrams have well-defined canonicity and regularization, which motivates us to implement quantum model-checking algorithms by means of DDs. Up to now, however, these quantum DDs have not been used in quantum model checking.

In this paper, we incorporate quantum DDs into quantum model checking for the first time. Quantum Markov chains (QMCs for short) have been adopted as a fundamental model of many quantum information processing systems (e.g. quantum communication protocols, semantics of quantum programs, etc). So, we choose to use QMCs as our system model. As is well-known, reachability analysis is a core task in classical model checking algorithms. In the quantum case, indeed, reachability analysis has been applied in quantum communication, quantum control and termination analysis of quantum programs among many others. Therefore, we focus on the issue of reachability analysis of quantum Markov chains. In addition, we decide to use Context-Free-Language Ordered Binary Decision Diagrams [15] (CFLOBDD for short), one of the most efficient quantum DDs as the backend of our tool to provide support for functionalities. We also refer to Quasimodo [14], a quantum circuit simulator based on CFLOBDD, for some of the code's implementation details. Supported by the efficiency of the DD representation, our tool is well scalable and has the potential to be expanded into large-scale quantum circuit model checkers in the future.

Contributions of the Paper: This paper introduces the first reachability analysis tool for QMCs, called QReach[1]. It can efficiently compute reachable subspaces of QMCs with the following techniques:

- Subspaces of QMCs, which are usually defined by atomic propositions in Birkhoff-von Neumman quantum logic, are represented as CFLOBDDs in QReach.
- Partitioning and frontier set simplification are introduced in our algorithm to reduce the size of data structures, in analogy to corresponding techniques in classical symbolic model checking.

2 Quantum Reachability Analysis

For convenience of the reader, we briefly review the model of QMCs and their reachable subspaces. Recall that a Markov chain (MC for short) is a pair $\langle S, P\rangle$, where S is a finite set of states and P is a transition probability matrix $P : S \times S \to [0,1]$ satisfying a normalization condition $\sum_{s' \in S} P(s,s') = 1$ for any $s \in S$. Similarly, a QMC is defined as a pair $\langle \mathcal{H}, \mathcal{E}\rangle$, where $\mathcal{H}$ is the state Hilbert

[1] Available at https://github.com/Acdimy/qreach-tools.

space of the quantum system under consideration and $\mathcal{E}$ is a quantum operation describing the evolution of the system, i.e. a mapping from a quantum state ρ to another $\mathcal{E}(\rho)$ (also called a quantum channel in quantum information literature). Table 1 gives a detailed comparison between classical MCs and QMCs:

Table 1. Classical Markov chains vs quantum Markov chains.

	(Discrete-time) Markov Chain	Quantum Markov Chain
State space	Finite or countable set	Finite-dimensional or separable Hilbert space
Initialization	Probability distribution: $\iota_{init} : S \to [0,1]$ $\sum_{s\in S} \iota_{init}(s) = 1$	Density matrix: $\rho = \sum_i p_i \lvert\psi_i\rangle\langle\psi_i\rvert$ $\sum p_i = 1$
Transition	Probability transition matrix: $P : S \times S \to [0,1]$ $\sum_{s'\in S} P(s,s') = 1$	Quantum operation: $\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$ $\sum E_i^\dagger E_i = I$
Logic	Probabilistic temporal logic	Temporal extension of Birkhoff-von Neumann quantum Logic

- A pure state of an n-dimensional quantum system is represented by a unit complex vector $|\psi\rangle \in \mathbb{C}^n$. In Table 1, the density operator $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ is a mathematical representation of ensemble $\{(p_i, |\psi_i\rangle\}$, meaning the system is in state $|\psi_i\rangle$ with probability p_i. Thus, ρ can be seen as a quantum analog of the initial probability distribution in a classical MC.
- The quantum operation $\mathcal{E}$ is a quantum generalization of the transition probability matrix in a classical MC. According to the principles of quantum mechanics, it can be mathematically modelled as $\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$, where each E_i is an $n \times n$ complex matrices (called *Kraus matrices*) satisfying the condition $\sum E_i^\dagger E_i = I$ (the unit matrix), which is a counterpart of the normalization condition in a classical MC. In particular, the evolution of a closed quantum system is described by $\mathcal{E}(\rho) = U\rho U^\dagger$, where U is a unitary matrix, i.e. $UU^\dagger = U^\dagger U = I$.

Quantum Reachability Problem: Given a QMC $\mathcal{C} = \langle \mathcal{H}, \mathcal{E}\rangle$ and any initial state ρ in $\mathcal{H}$, compute the reachable subspace:

$$\mathcal{R}_{\mathcal{C}}(\rho) = \text{span}\{|\psi\rangle \in \mathcal{H} : |\psi\rangle \text{ is reachable from } \rho \text{ in } \mathcal{C}\} \tag{1}$$

Intuitively, $\mathcal{R}_{\mathcal{C}}(\rho)$ consists of not only the states reached in the execution path $\rho \to \mathcal{E}(\rho) \to \mathcal{E}^2(\rho) \to \cdots \to \mathcal{E}^n(\rho) \to \cdots$ but also their linear combinations.

Example 1 (Quantum random walk). Consider a quantum random walk on a 4-cycle [16] with the Hadamard operator as a coin c, shown in Fig. 1. The walking

space is a 4-dimensional Hilbert space $\mathcal{H}_p$ on the bottom two qubits p_1, p_2, supported by four position states $\{|0\rangle_p, |1\rangle_p, |2\rangle_p, |3\rangle_p\}$ in the computational basis. After applying the coin H in each step, the evolution of the system is described by a quantum conditional (shift operator):

$$S = |0\rangle_c\langle 0| \otimes \sum_i |i+1\rangle_p\langle i| + |1\rangle_c\langle 1| \otimes \sum_i |i-1\rangle_p\langle i|$$

where the neighbor-state $|i+1\rangle$ and $|i-1\rangle$ are computed modulo 4. It means that the walker can simultaneously walk in different directions, which is the main difference between classic and quantum random walks.

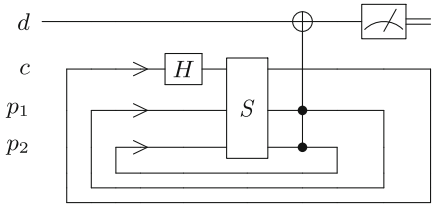

Fig. 1. Quantum random walk on a 4-cycle

This system can be modelled as a QMC with $\mathcal{E}$ being defined by the unitary operator $S(H_c \otimes I_p)$. Let it start in pure state $\rho = |000\rangle\langle 000|$. Using the tool QReach presented in this paper, one can compute that the reachable space of this QMC is the 6-dimensional space with linear independent basis $\{|000\rangle, |001\rangle + |111\rangle, |100\rangle - |110\rangle, |101\rangle + |001\rangle, |010\rangle, |011\rangle + |101\rangle\}$. It is surprising that not the whole 8-dimensional state space $\mathcal{H}_c \otimes \mathcal{H}_p$ is reachable although any position in $\mathcal{H}_p$ may be hit in some time.

3 Architecture and Data Structures

In this section, we elaborate on the architecture of QReach and some reasoning techniques for quantum circuits based on the CFLOBDD backend. Although our target is specified on quantum reachability analysis, we believe that some functionalities of QReach are also useful for other tasks.

3.1 Architecture of QReach

An overview of the architecture of QReach is presented in Fig. 3. For convenience of presentation, we describe QReach's procedure with Example 1. Let the system starting in state $\rho = |000\rangle\langle 000|$. To illustrate the capability of QReach for handling general quantum operations, we consider a faulty quantum random walk in which a bit-flip error happens in front of the Hadamard gate with probability p. The behavior of the system is then modelled by $\mathcal{E}(\rho) = E_0\rho E_0^\dagger + E_1\rho E_1^\dagger$,

```
# Prepare the BitFlipError
e = QError("Bitflip", loc=[0,0], params=[], p=0.5)

# Prepare the quantum Markov chain
qmc = QMarkov(cir_body="qrw.qasm", channels=[e])

# Prepare model checker and its CFLOBDD backend
qChecker = fromMarkovModel(qmc)
qChecker = initWithStr(qChecker, str_list=[])

# Execution and output
reachable_dim = qChecker.reachability()
qChecker.printProjector()
```

Fig. 2. A Demo for reachability analysis of QMCs. `channels` is a list of quantum errors and instructions like measurement and reset, containing their occurring positions.

where $E_0 = \sqrt{1-p}\ S(H_c \otimes I_p)$, and $E_1 = \sqrt{p}\ S(H_c \otimes I_p)(X \otimes I_p)$. Our purpose is to compute the reachable subspace of $\mathcal{E}$. An example Python program for this process is shown in Fig. 2.

We implement our symbolic quantum reachability analysis algorithm with CFLOBDD in a C++ core and provide Python interfaces for invoking. Some of the key components are explained below:

Input and Output. QReach accepts a composite input specification to represent a QMC. A quantum program written in the QASM format [7] is parsed as the main circuit body of the QMC. Some specified quantum errors, measurements, and other non-unitary channels can be coded as a supplement in the `Qchannel` type. Once results are obtained, some subspace characters (e.g. dimensions and support vectors) can be output.

Simulation. A well-formed toolkit Quasimodo [14] based on CFLOBDD was implemented for quantum algorithm simulation. Simulation for quantum circuits, or in other words, applying a sequence of quantum gates on a pure state, is an essential process during the reachability analysis. Therefore, we referred to some of the codes' implementation details from Quasimodo. Specifically, we adopted Quasimodo's Pybind architecture, which links C++ APIs and Python. However, a simulation framework like Quasimodo cannot handle situations in reachability analysis like mixed states and super-operators. We fixed these issues, added some new features, and stored gate sequences and intermediate projectors to make the simulation execution process not just sequential.

These methods are covered in `fromMarkovModel()` and `reachability()`, while still available to invoke independently for other purposes. For example, `qchecker.u3()` conducts normal U3 gate simulation on the state vector in the current workspace; `setProjector()` and `applyProjector()` methods provide data manipulation in *Projector* and *State vector* as shown in Fig. 3. Detailed techniques used in our CFLOBDD simulator different from that in previous works [14,15] will be discussed in Sect. 3.2.

Reachability Analysis. The efficient algorithm for quantum reachability analysis to be elaborated in Sect. 4 has been implemented in QReach. We also implemented the interfaces for some of mathematical tricks in [19] (e.g. Choi matrix transformation and maximally entangled state preparation) on CFLOBDDs, which will be critical in a future extension of QReach for computing reachability probabilities (rather than subspaces) of QMCs.

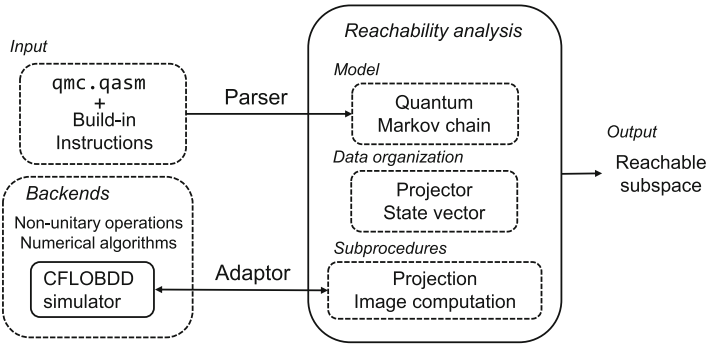

Fig. 3. Architecture of QReach.

3.2 CFLOBDD for Quantum Reachability Analysis

Now we introduce our CFLOBDD backend, which implements some support for numerical algorithms and quantum instructions. In particular, we illustrate how to expand simulation of a quantum system to its reachability analysis.

As a newly proposed DD-based structure, CFLOBDD attracts our attention due to its distinctive features compared to other DDs for quantum systems. CFLOBDD adopts a single-entry, multi-exit, non-recursive, hierarchical finite-state machine architecture [15]. From a programming perspective, a certain form of procedure call is invoked, leading to some exponential compression over BDDs. Following the name "context-free language", the incoming and outgoing edges of groupings are matched according to certain principles. Figure 4 provides a general insight into how the edges of CFLOBDDs are matched. The representing capability of the CFLOBDD is exploited in our tool QReach for symbolic reachability analysis of QMCs.

Like any other type of DDs, the compression capability of CFLOBDDs only stands out in specific instances. In these cases, canonical reduced forms reuse parts in a DD and save storage from the raw data. However, in general circumstances, the number of nodes required to represent a large-scale matrix or vector is still exponential. Reordering strategies for reduced ordered BDDs to optimize storage usage are hard (NP-complete) [3]. This problem becomes even more severe in algebraic DDs [1], where non-Boolean values make it harder to

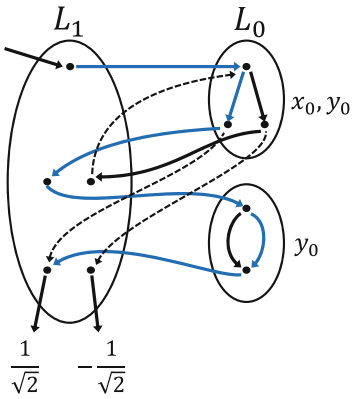

Fig. 4. CFLOBDD for Hadamard gate. Indices are represented in the L_0 groupings consisting of fork nodes and don't care nodes. A path from the entry of the topmost grouping to the terminal values denotes an assignment to all indices. For example, the bold blue path corresponds with the $\{x_0 = 0, y_0 = 1\}$ entry of the Hadamard matrix, resulting a value $\frac{1}{\sqrt{2}}$. (Color figure online)

find similar structures in a diagram. Therefore, we cannot simply represent a quantum operation or a projector as a single CFLOBDD without a partition strategy. Unlike classical symbolic model checking, a quantum circuit is usually difficult to partition due to entanglements. We chose an alternative in the QReach backend: using an augmented simulation method to calculate quantum operations. Thus, only state vectors and single quantum gates need to be stored, rather than the whole matrix.

In particular, circuits in QMCs are usually complicated, involving noises and dynamic operations. To handle them, we strengthen CFLOBDD with the following techniques:

Non-unitary Operators. Apart from normal quantum gates like Hadamard, Pauli, and generic U3 rotation gates, we specifically support two-dimensional matrices of any form, covering those non-unitary operators in quantum noises and measurements; for instance, amplitude damping channels with operators:

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \qquad E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}$$

Another example of non-unitary operators is Z-basis measurements. The post measurement states are obtained by applying $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ respectively, followed by normalizations.

Basic Methods Extensions. Some operations are added to CFLOBDD as a basis for top-level algorithms, incorporating the normalization and conjugate transpose. In addition to these methods, an optimizing trick for complex number representation is applied. We used a simplified version of the method proposed in [20], constructing a Hash function and unique table for complex numbers.

Numerical Algorithms. Numerical algorithms are critical in QReach. Based on them, some operations that are particularly important in modelling quantum systems (e.g. partial trace and Choi matrix) are now available in QReach (see Example 2). A key methodology is to decompose operands of calculations into base vectors, replacing complicated operations with matrix-vector multiplication or inner products of vectors. In Sect. 4, the high-level description of reachability analysis algorithm also embodies this idea.

Example 2 (Partial trace). Consider a quantum system composed of qubits A and B in state ρ_{AB}. Then the state of A can be described by the partial trace operator:

$$\rho_A \equiv tr_B(\rho_{AB}) := \sum_i (I_A \otimes \langle i|_B)\rho_{AB}(I_A \otimes |i\rangle_B)$$

For simplicity, suppose the system is in a pure state $|\psi\rangle = |0\rangle|\lambda\rangle + |1\rangle|\mu\rangle$. Tracing out qubit A, qubit B should be in the mixed state $\rho = |\lambda\rangle\langle\lambda| + |\mu\rangle\langle\mu|$. In QReach, this procedure is conducted in the following steps to avoid redundant matrix manipulations and adjustments to CFLOBDD's internal structures: (1) Perform a Z-basis measurement on A and get unnormalized post measurement states $|0\rangle|\lambda\rangle$ and $|1\rangle|\mu\rangle$; (2) Apply X gate to A conditionally on the measurement result one; (3) Let the collection $S = \{|0\rangle|\lambda\rangle, |0\rangle|\mu\rangle\}$. Then S can be viewed as ρ's representation and participate in later calculations. In some cases, like reachability analysis, the norm of a state vector is not essential and could be omitted thereby. Note that after these operations qubit A remains in a tensored zero state, because the number of qubits in a CFLOBDD is required to be an exponential power of 2. We apply an X gate on the measure-one result to make the effect looks like resetting a qubit.

4 Algorithm for Reachability Analysis

The existing algorithm for reachability analysis of QMCs is based on Choi matrix representation of quantum operations introduced in [19]. In this section, we propose a more efficient algorithm for the same purpose (see Algorithm 1).

Our algorithm is a natural extension of reachability analysis in classical model checking using a BFS-based technique. The main difference is that we are dealing with reachable *subspaces* of the Hilbert space $\mathcal{H}$ rather than *subsets* of a finite set of states in the classical case. Therefore, Algorithm 1 traverses each possible *dimension* of a finite-dimensional Hilbert space non-repetitively rather than each reachable state as in the classical case. Note that the code segment from Line 7 to Line 12 is the process of extracting vectors orthogonal to those that have been searched, which is similar to frontier set simplification in classical symbolic model checking.

The nontrivial subprocedures in Algorithm 1 differing from that in classical reachability analysis are the projection and image computation (Line 5 and Line 7). To reduce the representing and temporal cost in the algorithm, our basic idea

Algorithm 1. Computing reachable space

Input: Super operator $\mathcal{E}$ in Hilbert space $\mathcal{H}$ with dimension d; set of initial states P
Output: A set of orthogonal basis P' of reachable subspace of $\mathcal{H}$

$P' \leftarrow Gram\_Schmidt(P)$, $cnt \leftarrow size(P')$
Initialize queue Q with P'
while Q not empty and $cnt < d$ **do**
  $curr\_state \leftarrow Q$.pop()
  $expanded\_states \leftarrow$ `Image`$(\mathcal{E}, curr\_state)$
  for s in $expanded\_states$ **do**
    $s \leftarrow s -$ `Project`(P', s)
    $s \leftarrow$ `normalize`(s)
    if s is not zero vector **then**
      Q.push(s)
      P'.append(s)
      $cnt \leftarrow cnt + 1$
    end if
  end for
end while
return P'

is to take eigenvectors into calculation instead of the whole matrix. In practice, most of the projections are low-rank, which ensures the efficiency of this idea.

Implementation in QReach. For image computation, we exploit the simulation functionality of our backend data structure CFLOBDD. The runtime of the backend's simulation highly determines our algorithm's efficiency. In this step, non-unitary operations like noises, measurements, qubit resets, and deallocations will be simulated by the augmented simulator introduced in the past section. All the simulations of channels together make up the image computation of Kraus representations $\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger$.

A projector onto a subspace of the Hilbert space can be represented by a set of orthogonal support vectors of the subspace. This technique can be viewed as a quantum version of partitioning, which usually appears as forms of disjunctions and conjunctions in classical symbolic model checking [4]. Formally, let $\mathcal{P}$ be the projector onto a subspace with an orthonormal basis $\{|i\rangle\}$, that is, $\mathcal{P} = \sum |i\rangle\langle i|$, then we set P to be the set of $|i\rangle$'s, and

$$\texttt{Project}(P, |s\rangle) = \sum \langle s|i\rangle^* |i\rangle$$

We conduct conjugate-transposing on $|s\rangle$ instead of $|i\rangle$ to invoke vector operations as few as possible. The computational complexity of subprocedure `Image` and `Project` are both $O(d^2)$ given a constant number of Kraus matrices.

The following theorem shows the correctness and complexity of our algorithm.

Theorem 1. *The output P' and input P of Algorithm 1 satisfies: $span(P') = \mathcal{R}_{\mathcal{C}}(\rho) = \bigvee_{i=0}^{d-1} supp(\mathcal{E}^i(\rho))$, where ρ is the initial state, and $supp(\rho) = span(P)$. The time complexity of Algorithm 1 is $O(d^3)$*

Proof. Following the theorem 1 in [19], for $d = dim(\mathcal{H})$, and any density operator ρ in $\mathcal{H}$,

$$\mathcal{R}_{\mathcal{C}}(\rho) = \text{supp}\left(\sum_{i=0}^{d-1} \mathcal{E}^i(\rho)\right) \tag{2}$$

And all reachable states can be reached in at most d iterations. The correctness is proved in three steps:

i) The main *while* loop terminates in at most d steps;
ii) When terminates, $\text{span}(\texttt{Image}(\mathcal{E}, P')) = \text{span}(P')$;
iii) $\mathcal{R}_{\mathcal{C}}(P') = \mathcal{R}_{\mathcal{C}}(\rho)$.

At last, combining the complexity of broad-first-search and subprocedures, the Algorithm 1 has complexity $O(d^3)$, which is an improvement over $O(d^{4.7454})$ in [19]. □

The dimension of a Hilbert space often grows exponentially larger in quantum systems. Therefore, our algorithm will inevitably become inefficient on quantum circuits with more than twelve qubits. Although limited on the scale of quantum systems, Algorithm 1 is stable for the number of quantum operations and the dimension of initial spaces. The BFS strategy and the frontier set simplification ensure that only dimensions that are reached for the first time can be counted.

5 Use Cases and Experiments

Some cases are studied in this section, providing insight into practical applications of our tool QReach for quantum reachability analysis in the future. All experiments were conducted on a personal computer with hardware configurations: Intel i5-13600kf CPU with 14 cores; 32GB RAM. The experimental results are presented in Table 2.

Grover Search. The Grover search algorithm provides a quadratic speedup over a series of classical search algorithms [8]. The main idea of the Grover search is to apply a quantum subroutine iteratively which leads to a rotation from the initial state to the target state. QReach's reachability analysis under some float precision shows that during iterations, the state is always located in the 2-dimensional subspace spanned by the initial state and the target state.

Quantum Random Walk. We tested quantum random walk circuits (QRW) with different numbers of qubits which have a similar structure with Example 1. To demonstrate more functionalities of QReach, mixed initial states and amplitude dumping noises are introduced in front of the Hadamard gate. The (dimension of) reachable space computed by QReach for these quantum circuits are given in Table 2.

Table 2. Experimental results. The noise in QRW is amplitude dumping. For RUS, the circuits implement $(I+2iZ)/\sqrt{5}$, $(2X+\sqrt{2}Y+Z)/\sqrt{7}$, and $(3I+2iZ)/\sqrt{13}$ respectively.

	#Qubits	Ope. type	Initial dim.	Time(s)	#Edges	Reachable dim.
Grover	7	Unitary	1	0.0252	268	2
	15	Unitary	1	0.0303	350	2
	31	Unitary	1	0.0516	432	2
	63	Unitary	1	0.0885	514	2
QRW	3	Unitary	1	0.0284	435	6
	5	Unitary	1	0.0561	1337	10
	7	Unitary	1	0.196	7512	34
	9	Unitary	2	379.64	608097	512
	7	Noise	2	0.446	10211	64
	9	Noise	2	110.70	447811	512
	10	Noise	2	117.36	421038	1024
RUS	3	Measure	1	0.0262	130	2
	2	Measure	1	0.0216	74	2
	2	Measure	1	0.0203	60	1

Repeat-Until-Success Circuits. Repeat-until-success (RUS) circuits [13] are a type of circuit that decides whether to repeat or terminate based on the measurement results. It is usually used to design circuits with fewer non-Clifford gates or ancilla qubits. In QReach, it can be modelled as a quantum Markov chain with measurements and qubit resetting as parts of the channel. We tested some of the examples in [13]. It is clear that the reachable dimensions should be 2 or 1, depending on whether the resulting quantum states differ by only one global phase if the measurement succeeds or fails (Fig. 5).

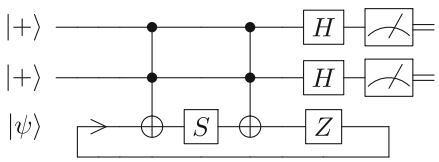

Fig. 5. A repeat-until-success circuit for gate $V_3 = (I + 2iZ)/\sqrt{5}$.

There is a consensus that every future tool released in quantum model checking must face the problem of finding broader applications. Besides these cases, we are improving the scope of QReach and exploring more possible applications on sequential quantum circuits and protocols.

References

1. Bahar, R.I., et al.: Algebric decision diagrams and their applications. Formal Methods Syst. Des. **10**, 171–206 (1997)
2. Baltazar, P., Chadha, R., Mateus, P.: Quantum computation tree logic-model checking and complete calculus. Int. J. Quantum Inform. **6**(02), 219–236 (2008)
3. Bollig, B., Wegener, I.: Improving the variable ordering of OBDDs is np-complete. IEEE Trans. Comput. **45**(9), 993–1002 (1996)
4. Burch, J.R., Clarke, E.M., Long, D.E., McMillan, K.L., Dill, D.L.: Symbolic model checking for sequential circuit verification. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **13**(4), 401–424 (1994)
5. Chen, Y.F., Chung, K.M., Lengál, O., Lin, J.A., Tsai, W.L.: AUTOQ: an automata-based quantum circuit verifier. In: Enea, C., Lal, A. (eds.) International Conference on Computer Aided Verification, pp. 139–153. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37709-9_7
6. Chen, Y.F., Chung, K.M., Lengál, O., Lin, J.A., Tsai, W.L., Yen, D.D.: An automata-based framework for verification and bug hunting in quantum circuits. Proc. ACM Program. Lang. **7**(PLDI), 1218–1243 (2023)
7. Cross, A., et al.: OpenQASM 3: a broader and deeper quantum assembly language. ACM Trans. Quantum Comput. **3**(3), 1–50 (2022)
8. Grover, L.K.: Quantum computers can search rapidly by using almost any transformation. Phys. Rev. Lett. **80**(19), 4329 (1998)
9. Hong, X., Zhou, X., Li, S., Feng, Y., Ying, M.: A tensor network based decision diagram for representation of quantum circuits. ACM Trans. Des. Autom. Electron. Syst. (TODAES) **27**(6), 1–30 (2022)
10. Mateus, P., Ramos, J., Sernadas, A., Sernadas, C.: Temporal logics for reasoning about quantum systems. Semantic Tech. Quantum Comput., 389–413 (2009)
11. Mateus, P., Sernadas, A.: Weakly complete axiomatization of exogenous quantum propositional logic. Inf. Comput. **204**(5), 771–794 (2006)
12. Niemann, P., Wille, R., Miller, D.M., Thornton, M.A., Drechsler, R.: QMDDs: efficient quantum function representation and manipulation. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(1), 86–99 (2015)
13. Paetznick, A., Svore, K.M.: Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries. Quantum Info. Comput. **14**(15–16), 1277–1301 (2014)
14. Sistla, M., Chaudhuri, S., Reps, T.: Symbolic quantum simulation with Quasimodo. In: International Conference on Computer Aided Verification. pp. 213–225. Springer (2023). https://doi.org/10.1007/978-3-031-37709-9_11
15. Sistla, M.A., Chaudhuri, S., Reps, T.: CFLOBDDs: context-free-language ordered binary decision diagrams. ACM Trans. Program. Lang. Syst. (2023)
16. Venegas-Andraca, S.E.: Quantum walks: a comprehensive review. Quantum Inf. Process. **11**(5), 1015–1106 (2012)
17. Vinkhuijzen, L., Coopmans, T., Elkouss, D., Dunjko, V., Laarman, A.: LIMDD: a decision diagram for simulation of quantum computing including stabilizer states. Quantum **7**, 1108 (2023)
18. Ying, M., Feng, Y.: Model Checking Quantum Systems: Principles and Algorithms. Cambridge University Press (2021)

19. Yu, N., Ying, M.: Reachability and termination analysis of concurrent quantum programs. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 69–83. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32940-1_7
20. Zulehner, A., Hillmich, S., Wille, R.: How to efficiently handle complex values? Implementing decision diagrams for quantum computing. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–7. IEEE (2019)