Equivalence Checking in Multi-level Quantum Systems

Philipp Niemann[1], Robert Wille[1,2], and Rolf Drechsler[1,2]

[1] Institute of Computer Science, University of Bremen, 28359 Bremen, Germany
[2] Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany
{pniemann,rwille,drechsle}@informatik.uni-bremen.de

Abstract. Motivated by its superiority compared to conventional solutions in many applications, quantum computation has intensely been investigated from a theoretical, physical, and design perspective. While these investigations mainly focused on two-level quantum systems, recently also advantages and benefits of higher-level quantum systems became evident. Though this led to several approaches for the representation and realization of quantum functionality in different dimensions, no efficient solution for verifying their equivalence has been proposed yet. In the present paper, we address this problem. We propose a scheme which is capable of verifying the equivalence of two quantum operations regardless of the dimension of their underlying quantum system. The proposed scheme can be incorporated into data-structures such as *Quantum Multiple-Valued Decision Diagrams* (QMDD) particularly suited for the representation of quantum functionality and, by this, enables an efficient verification. Experiments confirm the efficiency of the proposed approach.

1 Introduction

Quantum computation [19] provides a new way of computation based on so called *qubits*. In contrast to conventional bits, qubits do not only allow to represent the (Boolean) basis states 0 and 1, but also superpositions of both. By this, qubits can represent multiple states at the same time which enables massive parallelism. Additionally exploiting further quantum mechanical phenomena such as phase shifts or entanglement enables asymptotic speed-ups for many relevant problems (e.g. database search or integer factorization), offers new methods for secure communication (e.g. quantum key distribution), and has several other appealing applications [19].

Motivated by these prospects, researchers from various domains investigated this emerging technology. While, originally, the exploitation of quantum mechanical phenomena has been discussed in a purely theoretical fashion (see e.g. [10,23] for two well-known quantum algorithms), recently also the consideration of physical realizations (see e.g. [6,8,21]) as well as proper design methods (see e.g. [1]) gained significant interest. However, most of these considerations and implementations focused on two-level quantum systems, i.e. systems based on qubits.

But, as a matter of fact, the considered quantum systems offer multiple levels to be exploited. These levels are readily accessible and using them for state preparation and read-out has been demonstrated [18]. By this, computations can be performed on so called *qudits* rather than qubits. Researchers investigated possible exploitations of these additional levels e.g. for matters of simplified implementation or improved design of quantum operations. They were able to show that multi-level systems are useful for many promising applications and provide several practical advantages in the design of respective operations (see e.g. [5, 12]). This is discussed in detail later in Section 3.

As a consequence, several approaches for representing and realizing quantum functionality in various quantum systems exist. This raises the question of how to verify whether or not two quantum operations given in different quantum systems indeed realize the same function. Although several methods for equivalence checking of quantum functionality have been proposed in the past (e.g. based on simulation [24], decision diagrams [26], or Boolean satisfiability [28]), all of them only supported two-level quantum systems composed of qubits.

In this work, we address the problem of checking functional equivalence between operations that are realized in multi-level quantum systems. This explicitly includes comparisons between realizations in different dimensions, i.e. quantum systems with a different number of levels. For this purpose, we first discuss and define functional equivalence in this context. Afterwards, a verification scheme based on the formal representation of quantum operations by unitary matrices is proposed. Since these matrices grow exponentially with the number of considered qubits, we additionally demonstrate how the proposed scheme can be incorporated into data-structures such as QMDDs [15] which are explicitly suited for the compact representation of quantum operations. By this, an equivalence checker for multi-level quantum systems results. The efficiency of the proposed scheme is confirmed by an experimental evaluation considering a wide range of operations realized in different quantum systems.

The remainder of the paper is structured as follows. In Section 2, preliminaries on quantum computation as well as a proper data-structure for the compact representation of quantum functionality are briefly reviewed. Section 3 discusses recent achievements in the field of multi-level quantum systems and, by this, motivates the present work. A definition of functional equivalence in multi-level quantum systems is then provided in Section 4 before the proposed scheme and an efficient implementation are described in detail. The paper concludes with a summary on the conducted experimental evaluation in Section 5 and our conclusions in Section 6.

2 Preliminaries

This section briefly reviews the basics on quantum computation. Furthermore, we sketch the main ideas of *Quantum Multiple-valued Decision Diagrams (QMDDs)*, a data-structure which is used later for an efficient implementation of the proposed equivalence checking scheme.

2.1 Quantum Computation

Most commonly, the basic building blocks for quantum computation are qubits. A *qubit* is a two-level quantum system, described by a two-dimensional complex Hilbert space. The two orthogonal *basis states* $|0\rangle \equiv \binom{1}{0}$ and $|1\rangle \equiv \binom{0}{1}$ are used to represent the (conventional) values 0 and 1. Any state of a qubit may be written as $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers with $|\alpha|^2 + |\beta|^2 = 1$. The quantum state of a single qubit is denoted by the vector $\binom{\alpha}{\beta}$. We say that a qubit is in *superposition* if neither of the so called *amplitudes* α or β is zero. A qubit can be *measured*, yielding either the result $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ or $|\beta|^2$, respectively. Such measurement destroys superposition and forces the qubit to the respective basis state. The state of a quantum system with $n > 1$ qubits is given by an element of the tensor product of the single qubit spaces, i.e. a linear combination of the *tensor states* $|0\ldots0\rangle, |0\ldots1\rangle, \ldots, |1\ldots1\rangle$, which are the tensor products of basis states. Consequently, a quantum state is represented as a normalized vector of length 2^n (called the *state vector*), whose components denote the amplitude for each tensor state.

By the postulates of quantum mechanics, the evolution of a quantum system due to a quantum operation can be described by a *unitary transformation matrix* U [19]. Here, the columns correspond to the output state vectors that result when applying the respective operation to the tensor states as inputs. Thus, the entry u_{ij} of the matrix describes the mapping from the input tensor state $|j\rangle$ to the output tensor state $|i\rangle$.

Example 1. Commonly used quantum operations include the *Hadamard* operation H (setting a qubit into a balanced superposition) and the T (or $\frac{\pi}{8}$) operation. The corresponding unitary matrices are defined as

$$H = \tfrac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix}1 & 0\\ 0 & e^{\frac{\pi i}{8}}\end{pmatrix}.$$

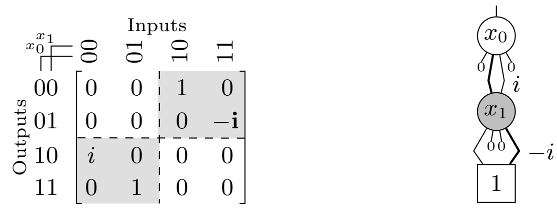
Applying these operations to a qubit in basis state $|1\rangle$ yields

$$H|1\rangle = \tfrac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \tfrac{1}{\sqrt{2}}\begin{pmatrix}1\\ -1\end{pmatrix} = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \;\; \text{and}$$

$$T|1\rangle = \begin{pmatrix}1 & 0\\ 0 & e^{\frac{\pi i}{8}}\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \begin{pmatrix}0\\ e^{\frac{\pi i}{8}}\end{pmatrix} = e^{\frac{\pi i}{8}}|1\rangle, \text{ respectively.}$$

While these operations work on a single qubit, there are also operations on multiple qubits. Usually, these are *controlled* operations in the sense that the state of the additional *control qubits* determines which operation is performed on the *target qubit*.

Example 2. An important example of a controlled operation is the *controlled NOT* (CNOT) which flips the two basis states of the target qubit if and only if the control qubit is in the $|1\rangle$-state.



(a) Matrix. (b) QMDD.

Fig. 1. Matrix and QMDD representation of a 2-qubit quantum operation.

It has been shown that the set of CNOT, H, and T operations (forming the so-called *Clifford+T library*) is *universal* for quantum computation, i.e. operations from this set can approximate every unitary transformation to an arbitrary precision [3]. Moreover, these quantum operations can be implemented in a fault-tolerant fashion [3] – a crucial property since quantum computing is inherently very sensitive to environmental factors such as radiation and, hence, fault-tolerance is even more important than for conventional systems.

2.2 Quantum Multiple-valued Decision Diagrams

QMDDs [15] have been introduced as a data-structure for the efficient representation and manipulation of quantum operations. The main idea is a recursive partitioning of the respective transformation matrix and the use of edge and vertex weights to represent various complex-valued matrix entries. More precisely, a transformation matrix of dimension $r^n \times r^n$ is successively partitioned into r^2 sub-matrices of dimension $r^{n-1} \times r^{n-1}$. This partitioning is represented by a directed acyclic graph – the QMDD. The following example illustrates main aspects of this data-structure.

Example 3. Figure 1a shows a transformation matrix for which a QMDD as shown in Fig. 1b has been built. Here, the unique *root vertex* (labelled x_0) represents the whole matrix and has four outgoing edges to vertices representing the top-left, top-right, bottom-left, and bottom-right sub-matrix (from left to right). This decomposition is repeated at each partitioning level until the terminal vertex (representing a single matrix entry) is reached. To obtain the value of a particular matrix entry, one has to follow the corresponding path from the root vertex at the top to the terminal vertex and multiply all edge weights on this path. For example, the matrix entry $-i$ from the top right sub-matrix of Fig. 1a (highlighted bold) can be determined as the product of the weights on the highlighted path of the QMDD in Fig. 1b. For simplicity, we omit edge weights equal to 1 and indicate edges with a weight of 0 by stubs.

QMDDs are canonical representations, if normalization of edge weights (as described in [15]) is performed. Thus, they are very convenient for equivalence checking. Indeed, due to standard decision diagram techniques like *unique tables*, this task can be performed in $\mathcal{O}(1)$ by comparing root vertices.

3 Motivation: Multi-level Quantum Systems

Research on quantum computation is considered in numerous facets. Originally, the exploitation of quantum mechanical phenomena e.g. for data-base search [10], factorization [23], and other applications has been discussed in a purely theoretical fashion. But in the past decade also several physical realizations have been proposed – including prototypical implementations based on trapped ions [6], photons [21], and superconducting qubits [8]. However, most of these considerations and implementations focused on two-level quantum systems, i.e. systems based on qubits with the basis states $|0\rangle$ and $|1\rangle$ as reviewed in Section 2.1.

But, as a matter of fact, quantum computation allows for multiple basis states. Instead of qubits, *d-leveled qudits* are then used as basic building blocks. These do not rely on only two orthogonal basis states but a total of d basis states $|0\rangle, |1\rangle, \ldots, |d-1\rangle$. More precisely, a qudit is described by a d-dimensional Hilbert space, where the *state space* is formed by all superpositions $|\Psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$ for complex-valued α_i with $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$. Prominent examples of qudits are *qutrits* ($d = 3$) and *ququarts* ($d = 4$) which received most attention so far [5, 9, 11, 13, 16].

Multiple qudits with levels $d_0, \ldots, d_{n-1}$ form a *$\hat{d}$-level quantum system* where $\hat{d}$ is the maximum of the d_i. The underlying Hilbert space is the tensor product of the respective spaces of the single qudits. Accordingly, the state of such systems can be expressed by a state vector of length $\prod_{i=0}^{n-1} d_i$ and is given by a linear combination of the tensor states $|x_0, \ldots, x_{n-1}\rangle$ where $0 \leq x_i < d_i$ for $0 \leq i < n$.

Operations over qudits are described by extended unitary transformation matrices.

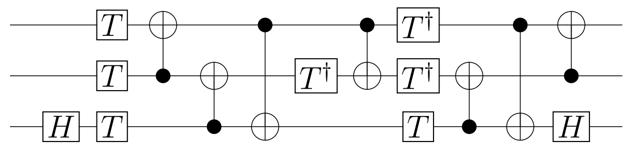
Example 4. The qutrit operation X which exchanges the basis states $|0\rangle$ and $|2\rangle$ can be described by the matrix

$$X_{0,2} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \text{ while } H_{0,1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{pmatrix}$$

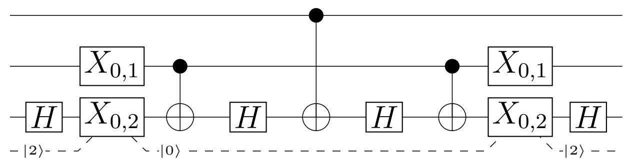
represents the ququart operation that performs the Hadamard operation on basis states $|0\rangle$ and $|1\rangle$, leaving the remaining basis states untouched.

Multi-level systems are not only of theoretical interest [9], but are also useful for promising applications of quantum computation (see e.g. [5, 12]). Moreover, the use of multi-level quantum systems offers several practical advantages compared to qubit systems. More precisely:

- Multi-level quantum systems allow for much more efficient realizations of multi-qubit operations [12]. For example, Fig. 2a shows a minimal implementation (in terms of T-depth, i.e. the number of sequential T operations) of a Toffoli operation within the Clifford+T library, i.e. based on a two-level



(a) Using the (two-level) Clifford+T library.



(b) Using a multi-level system.

Fig. 2. Realizations of the Toffoli operation.

system (taken from [1])[3]. The same functionality can be realized with significantly less operations in a multi-level system using a qutrit as shown in Fig. 2b (taken from [12]).

– A theoretical analysis showed that ququart operations may have a general advantage over qubit operations when it comes to the realization of generalized Toffoli operations. In fact, mapping these Toffoli operations to quantum operations using qubit-based techniques (e.g. [2]) requires an exponential effort. In contrast, a recently proposed four-valued approach can realize each Toffoli operation with linear complexity [22].

These advantages lead to an increased interest in multi-level quantum systems and the implementation of quantum operations in various dimensions. Consequently, as for qubit systems, the synthesis of general quantum functionality has also been studied for multi-level systems [4,7,17]. In [7], a generalized CNOT operation is suggested that reacts on an arbitrary control state and swaps an arbitrary pair of states on the target qudit. The advantage of this approach is that it is physically realizable by using standard CNOT operations and certain laser beams (Rabi oscillations) to swap basis states. By this, synthesis of many important multi-level circuits becomes possible with established technology.

Overall, various representations and realizations of quantum functionality for different quantum systems exist. But whether or not two given quantum operations in different dimensions indeed realize the same functionality has hardly been considered yet. This issue is addressed in the following, i.e. we present a scheme which automatically checks for the equivalence of operations in multi-level quantum systems.

[3] As established in the literature, horizontal lines represent the qudits and the operations $\boxed{H}$, $\boxed{T}$, $\bullet\!\!-\!\!\oplus$ (CNOT), etc. are applied successively from left to right.

4 Equivalence Checking in Multi-level Quantum Systems

While, thus far, equivalence checking for quantum functionality has intensely been considered in the past (leading to approaches e.g. based on simulation [24], decision diagrams [26], or Boolean satisfiability [28]), usually only operations in the same dimension have been compared. In this work, we propose a verification scheme which is capable of proving the functional equivalence between quantum operations even if they are realized in different dimensions. For this purpose, this section first discusses fundamental preconditions and provides a precise definition of the functional equivalence that we are going to address. Afterwards, the proposed equivalence checking scheme is introduced. Based on these concepts, we finally illustrate an efficient implementation of the proposed scheme.

4.1 Functional Equivalence for Quantum Operations

The purpose of equivalence checking is to verify whether two quantum operations realize the same functionality. In the following, we denote the two quantum operations to be compared by U_1 and U_2. The underlying quantum systems may have different dimensions d_1 and d_2 (for U_1 and U_2, respectively), where we assume $d_2 \geq d_1$ (without loss of generality). In order to check for equivalence between U_1 and U_2, it is important to have a precise definition of which basis states of the quantum systems actually correspond to each other. Basis states can either be *shared states*, if there is a corresponding basis state in the other system, or *don't care states*, if there is no counterpart.

Example 5. Consider two quantum operations U_1 and U_2, which are realized in a 2-level and 3-level quantum system, respectively. More precisely, the 2-level system consists of three qubits whereas the 3-level system is a hybrid system composed of two qubits and a single qutrit. A possible mapping between basis states is shown in Fig. 3. Here, all basis states are shared states except the $|1\rangle$ state of the qutrit in U_2, which has no counterpart in U_1 and, thus, is a don't care state.

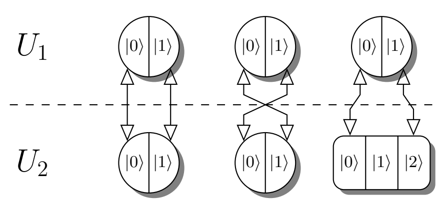


Fig. 3. Possible mapping of basis states between quantum systems.

In the following, the correspondence of basis states is represented by a function ψ. It is assumed that ψ is either derived from the specification of the respective technology mapping or directly provided by the designer.

In this work, we require that both quantum systems are composed of the same number of qudits and do not consider corner cases in which e.g. a ququart is realized by two qubits or even more scattered mappings. Although the proposed approach could be extended in order to support also these cases, our simplification is strongly motivated by the following facts:

- It is a natural requirement to enable the same set of measurements for U_1 and U_2. Since only entire qudits can be measured, this is only possible if there is a one-to-one relation between qudits in both systems.
- In order to interpret a measurement result correctly, there may not be cross-mappings between basis states that do not belong to corresponding qudits.

Don't care states may be employed during the operation, like e.g. in the multi-level realization of the Toffoli operation shown in Fig. 2b. But, we assume that neither input nor corresponding output states carry a don't care component.

Having these definitions and assumptions, two quantum operations U_1 and U_2 are *functionally equivalent* ($U_1 \equiv U_2$) if they perform an equivalent transformation on shared states. The behaviour on don't care states, however, may be arbitrary.

Example 6. Consider the matrix $H_{0,1}$ from Example 4 describing a Hadamard operation on a ququart. Assuming the trivial mapping of shared states $\psi(|i\rangle) = |i\rangle$ (for $i = 0, 1$), $H_{0,1}$ is equivalent to the Hadamard operation H on a qubit (from Example 1). However, with the same mapping, this is not the case for

$$H_{0,2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{pmatrix},$$

which also performs a Hadamard operation on a ququart, but on different basis states.

4.2 Proposed Equivalence Checking Scheme

Assume two quantum operations U_1 and U_2 (realized in quantum systems with dimensions $d_2 \geq d_1$) together with a mapping ψ and the corresponding distinction in shared states and don't care states. Then, functional equivalence of these operations can be verified in two steps:

1. Check whether the sub-matrices of U_1 and U_2 representing the mapping of shared input states to shared output states are equivalent.
2. Check whether the sub-matrices of U_1 and U_2 representing the mapping of don't care input states to shared output states (and vice versa) are zero matrices.

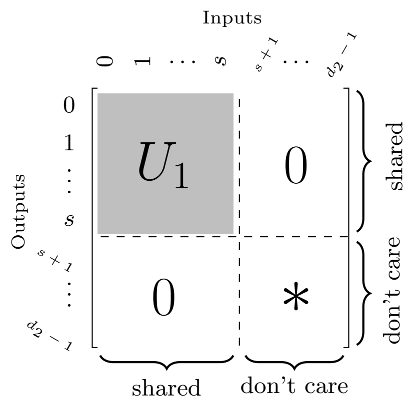


Fig. 4. Matrix of U_2 to be compared against U_1.

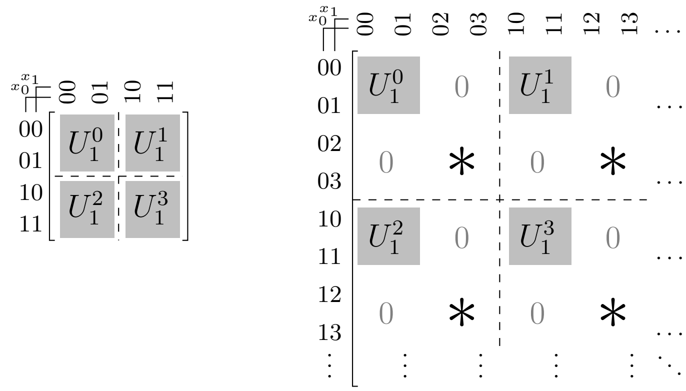
If both checks evaluate to true, then U_1 and U_2 are equivalent. This scheme is illustrated by means of Fig. 4 on the basis of single qudit systems. More precisely, Fig. 4 shows the matrix representing the quantum operation U_2, i.e. within the higher level system. Without loss of generality, assume that the basis states $|0\rangle, \ldots, |s\rangle$ of the U_1-system are shared states ($s < d_1$) and that ψ maps them to the basis states $|0\rangle, \ldots, |s\rangle$ of the U_2-system. The remaining states are assumed to be don't cares. Then, the top-left $(s+1) \times (s+1)$ sub-matrix of U_2 in Fig. 4 represents the mapping of shared input to shared output states. If $U_1 \equiv U_2$, this mapping obviously has to be equivalent to the corresponding mapping described in U_1. This is checked in Step 1.

Next, we exploit the fact that, as discussed in Section 4.1, only superpositions of shared basis states are applied to U_2, i.e. the basis states $|s+1\rangle, \ldots, |d_i - 1\rangle$ are always prepared (expected) with zero amplitude for input (output) states. Because of that and in order to keep the unitarity of the overall matrix, no further mappings from don't care states to shared states (represented in the top-right sub-matrix) and from shared states to don't care states (represented in the bottom-left sub-matrix) must exist. That is, the corresponding matrices have to be zero matrices. This is checked in Step 2. Note that we do not need to consider the bottom-right sub-matrix representing the mapping from don't care input to don't care output states, since arbitrary behaviour is allowed here.

Example 7. Once again, consider the operations H (from Example 1) and $H_{0,1}$ (from Example 4) together with the trivial mapping of shared states between the underlying 2- and 4-level quantum systems (i.e. $\psi(|i\rangle) = |i\rangle$ for $i = 0, 1$).

The 4-level operation $H_{0,1}$ is equivalent to the 2-level operation H, because (1) the mappings of shared states are equivalent and (2) no mappings from don't care states to shared states and vice versa exist. In contrast, these properties do not hold for the operation $H_{0,2}$ (from Example 6), showing its non-equivalence to the other two operations.

This scheme can accordingly be extended to quantum systems composed of an arbitrary number of qudits. Then, however, the checks have to consider the



(a) U_1 (2-level system). (b) U_2 (4-level system).

Fig. 5. Equivalence of operations in multi-qudit systems.

more scattered distribution of the respective sub-matrices. This is sketched in Fig. 5, where U_1 (realized in a 2-level quantum system) is to be compared to U_2 (realized in a 4-level quantum system composed of two ququarts). Here we assume that there are no don't care states in the U_1-system and again, without loss of generality, that ψ maps the basis states $|0\rangle$ and $|1\rangle$ of the U_1-system to the shared basis states $|0\rangle$ and $|1\rangle$ of the U_2-system. As can be seen, all (shared and don't care) basis states are considered separately for each qudit. Accordingly, the sub-matrices to be checked against U_1, the zero matrices, and don't care matrices ($*$) are scattered throughout the whole transformation matrix.

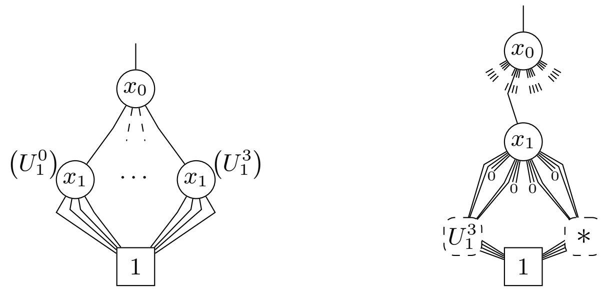
This, however, does not restrict the applicability of the proposed equivalence checking scheme, but of course harms the efficiency of the checks. Note that this is even more the case for more complex mappings of shared states. Then, the matrices under consideration can be in a more dispersed shape and the scheme might result in checking equivalence of many small non-adjacent sub-matrices.

Hence, an efficient implementation of this scheme even in these cases is essential and will be described next.

4.3 Implementation Using QMDDs

While the concepts introduced above are sufficient to check equivalence between arbitrary quantum operations, the matrix representations used thus far constitute a serious hurdle to the applicability of the proposed scheme. In fact, matrix descriptions grow exponentially with the number of qudits in a system. Hence, a naive implementation based on matrices is infeasible for quantum systems of a certain size.

In order to address this issue, we implemented the proposed scheme by means of the QMDD data-structure introduced in [15]. In this data-structure, each vertex represents a matrix which is partitioned into four sub-matrices (for qubit systems). Each sub-matrix is then represented by a successor of the current vertex. In case of multi-level quantum systems, the number of successors grows accordingly with the number of basis states.



(a) QMDD for U_1 (b) QMDD for U_2

Fig. 6. QMDD representations of the quantum operations sketched in Fig. 5.

Example 8. Figure 6 sketches the QMDD representations of the quantum operations already discussed in Fig. 5. As U_1 assumes a two-level quantum system, the overall matrix is partitioned into four sub-matrices. In contrast, the four-level system of U_2 is composed of $4 \cdot 4 = 16$ sub-matrices. Hence, the respective nodes have four and 16 successors, respectively. The x_1-vertices in Fig. 6a represent the sub-matrices U_1^0 and U_1^3, respectively (as indicated in brackets). The x_1-vertex in Fig. 6b sketches the second-top-right sub-matrix. Its sub-blocks U_1^3 and $*$ are represented by distinct sets of edges (which are indicated by a correspondingly labelled $\boxed{\text{M}}$, but are not part of the original QMDD).

Due to efficient techniques like *shared nodes* or *unique tables*, QMDDs are capable of representing quantum functionality for several dozens of qubits and/or qudits. Moreover, computed tables enable a very efficient implementation of the equivalence checking scheme outlined above.

For the purpose of equivalence checking, the QMDD representations of the operations have to be aligned. More precisely, we

- align the number of don't care states for corresponding qudits by "blowing up" vertices with additional successors (e.g. to introduce two additional don't care states for each qubit, all vertices in Fig. 6a are equipped with 12 additional 0-edges),
- align basis states (if the mapping of shared states is non-trivial) by rearranging edges appropriately, and
- align possibly different don't care to don't care mappings ($*$) by setting the corresponding edges to zero.

This transformation can be done in a single traversal of each QMDD and leads to representations of two matrices (of equal size), which are identical if and only if the operations are functionally equivalent. The latter can be verified in constant time by a single unique table look-up, since QMDDs provide canonical representations. By this, equivalence checking can be conducted efficiently even for larger quantum systems. This has been confirmed by an experimental evaluation whose results are summarized and discussed in the next section.

5 Experimental Results

The equivalence checking scheme described above has been implemented in C on top of the original QMDD package presented in [15][4] and evaluated on a wide range of operations realized in different quantum systems. More precisely, we considered

- 2-level and 4-level representations of various quantum operations including Shor's 9-qubit error correcting code (denoted by *9qubitN1* and *9qubitN2*), as well as a 7-qubit encoding (denoted by *7qubitcode*) taken from [14] and instances of Grover's algorithm (denoted by *Grover-k*) and quantum Fourier transforms (denoted by *QFT-k*) taken from [19] (k is the number of qubits),
- multi-qubit operations taken from RevLib [27], mainly realizing Boolean functionality for 2-level systems that additionally have been mapped to 4-level representations based on the methods described in [22] (denoted by their respective RevLib identifier), and
- randomly generated quantum operations with up to 25 qubits (denoted *arbitrary*).

In total, 296 benchmarks have been considered. For each of them, the 2-level representation has been compared against the respective 4-level representation. In order to additionally evaluate the performance of the proposed approach for non-equivalent operations, for each pair of representations we introduced an error through random changes (to one of them) and compared this to the original operation. All experiments have been conducted on a 2.8 GHz Intel Core i7 machine with 8 GB of main memory running Linux. The timeout was set to 500 CPU seconds.

The results are summarized in Table 1 for a selection of the conducted experiments[5]. The first two columns provide the identifiers of the respective benchmarks followed by its number of qudits. Afterwards, the run-time (in CPU seconds) for building up the data-structure (QMDD) as well as performing the actual equivalence check (EC) is provided for both cases, i.e. when both operations are equivalent and when they are not equivalent. As can be seen, the proposed scheme is able to efficiently check the equivalence of two quantum operations for the majority of all benchmarks. In fact, for 224 out of the 296 benchmarks, we were able to check their equivalence in less than a minute. While the actual equivalence check can always be conducted in almost no time, the limiting factor is the time needed for the construction of the representation of the respective quantum functionality, i.e. the QMDD in this case. Hence, the efficiency of the proposed scheme only relies on the chosen description mean. As improving those is an active research area (see e.g. the work on alternative representations such as *XQDDs* [26], *QuIDDs* [25] or improvements on QMDDs themselves [20]) and the proposed scheme can easily be adapted to other representations, further benefits can be expected here in the future.

[4] We thank the authors of [15] for providing us with their implementation of the QMDD package.

[5] Due to space limitations, we were not able to provide the numbers for all benchmarks.

Table 1. Experimental evaluation

Benchmark	#Qudits	Runtimes (s) Equivalence QMDD	EC	Non-Equivalence QMDD	EC
7qbitcode	7	< 0.01	< 0.01	< 0.01	< 0.01
9qubitN1	9	< 0.01	< 0.01	< 0.01	< 0.01
9qubitN2	17	0.04	< 0.01	0.04	< 0.01
Grover-5	11	0.41	< 0.01	0.38	< 0.01
Grover-6	13	0.04	< 0.01	0.05	< 0.01
QFT-5	5	< 0.01	< 0.01	0.01	< 0.01
QFT-7	7	0.01	0.01	0.02	< 0.01
add16_174	49	0.03	< 0.01	0.02	< 0.01
add32_183	97	0.08	< 0.01	0.08	< 0.01
alu2_199	16	117.84	0.01	115.94	0.02
alu3_200	18	224.42	0.04	217.3	0.04
apla_203	22	14.77	0.02	15.3	0.02
bw_291	87	> 500	–	> 500	–
cm163a_213	29	1.63	< 0.01	1.74	0.03
cu_219	25	4.36	< 0.01	4.59	0.02
cycle10_293	39	22.91	< 0.01	25.29	< 0.01
ham15_107	15	103.77	0.31	88.6	0.25
hwb7_61	7	3.24	< 0.01	2.94	< 0.01
lu_326	299	> 500	–	> 500	–
mod5add_306	32	326.98	0.4	307.95	0.36
arbitrary10	10	0.7	< 0.01	0.73	< 0.01
arbitrary15	15	15.04	0.2	25.41	0.55
arbitrary20	20	26.76	0.15	41.34	0.35
arbitrary25	25	> 500	–	> 500	–

6 Conclusions

In this work, we presented a scheme for checking the equivalence between two quantum operations working in different quantum systems. By this, the recent developments showing the advantages and benefits of multi-level quantum systems are taken into account. The proposed scheme can be incorporated into data-structures particularly suited for the representation of quantum functionality. An experimental evaluation confirmed that this enabled an efficient and fast equivalence checking which is mainly limited by the representation of the applied quantum functionality.

References

1. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans. on CAD 32(6), 818–830 (2013)
2. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Physical Review A 52(5), 3457–3467 (1995)
3. Boykin, P.O., Mor, T., Pulver, M., Roychowdhury, V., Vatan, F.: A new universal and fault-tolerant quantum basis. Information Processing Letters 75(3), 101–107 (2000)
4. Bullock, S.S., O'Leary, D.P., Brennen, G.K.: Asymptotically optimal quantum circuits for d-level systems. Physical Review Letters 94(23), 230502 (2005)
5. Cabello, A., D'Ambrosio, V., Nagali, E., Sciarrino, F.: Hybrid ququart-encoded quantum cryptography protected by Kochen-Specker contextuality. Physical Review A 84(3), 030302 (2011)

6. Cirac, J.I., Zoller, P.: Quantum computations with cold trapped ions. Physical Review Letters 74(20), 4091–4094 (1995)
7. Di, Y.M., Wei, H.R.: Synthesis of multivalued quantum logic circuits by elementary gates. Physical Review A 87, 012325 (2013) (2013)
8. Galiautdinov, A.: Generation of high-fidelity controlled-not logic gates by coupled superconducting qubits. Physical Review A 75(5), 052303 (2007)
9. Greentree, A.D., Schirmer, S., Green, F., Hollenberg, L.C., Hamilton, A., Clark, R.: Maximizing the Hilbert space for a finite number of distinguishable quantum states. Physical Review Letters 92(9), 097901 (2004)
10. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Theory of computing. pp. 212–219 (1996)
11. Klimov, A., Guzman, R., Retamal, J., Saavedra, C.: Qutrit quantum computer with trapped ions. Physical Review A 67(6), 062313 (2003)
12. Lanyon, B.P., Barbieri, M., Almeida, M.P., Jennewein, T., Ralph, T.C., Resch, K.J., Pryde, G.J., OBrien, J.L., Gilchrist, A., White, A.G.: Simplifying quantum logic using higher-dimensional Hilbert spaces. Nature Physics 5(2), 134–140 (2008)
13. Mc Hugh, D., Twamley, J.: Trapped-ion qutrit spin molecule quantum computer. New Journal of Physics 7(1), 174 (2005)
14. Mermin, N.D.: Quantum Computer Science: An Introduction. Cambridge University Press (2007)
15. Miller, D.M., Thornton, M.A.: QMDD: A decision diagram structure for reversible and quantum circuits. In: Int'l Symp. on Multi-Valued Logic. p. 30 (2006)
16. Moreva, E., Maslennikov, G., Straupe, S., Kulik, S.: Realization of four-level qudits using biphotons. Physical Review Letters 97(2), 023602 (2006)
17. Muthukrishnan, A., Stroud Jr, C.: Multivalued logic gates for quantum computation. Physical Review A 62(5), 052309 (2000)
18. Neeley, M., Ansmann, M., Bialczak, R.C., Hofheinz, M., Lucero, E., O'Connell, A.D., Sank, D., Wang, H., Wenner, J., Cleland, A.N., et al.: Emulation of a quantum spin with a superconducting phase qudit. Science 325(5941), 722–725 (2009)
19. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge Univ. Press (2000)
20. Niemann, P., Wille, R., Drechsler, R.: On the "Q" in QMDDs: Efficient representation of quantum functionality in the QMDD data-structure. In: RC. pp. 125–140 (2013)
21. O'Brien, J.L., Akira Furusawa, J.V.: Photonic quantum technologies. Nature Photonics 3(12), 687–695 (2009)
22. Sasanian, Z., Wille, R., Miller, D.M.: Realizing reversible circuits using a new class of quantum gates. In: Design Automation Conf. pp. 36–41 (2012)
23. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. Foundations of Computer Science pp. 124–134 (1994)
24. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Checking equivalence of quantum circuits and states. In: Int'l Conf. on CAD. pp. 69–74 (2007)
25. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Quantum Circuit Simulation. Springer, Dordrecht, Heidelberg, London, New York (Dec 2009)
26. Wang, S.A., Lu, C.Y., Tsai, I.M., Kuo, S.Y.: An XQDD-based verification method for quantum circuits. IEICE Transactions 91-A(2), 584–594 (2008)
27. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: an online resource for reversible functions and reversible circuits. In: Int'l Symp. on Multi-Valued Logic. pp. 220–225 (2008), RevLib is available at http://www.revlib.org
28. Yamashita, S., Markov, I.L.: Fast equivalence-checking for quantum circuits. Quantum Information & Computation 10(9&10), 721–734 (2010)