Overcoming the Trade-off between Accuracy and Compactness in Decision Diagrams for Quantum Computation

Philipp Niemann, *Member, IEEE*, Alwin Zulehner, *Student Member, IEEE*,
Rolf Drechsler *Fellow, IEEE*, Robert Wille, *Senior Member, IEEE*

***Abstract*—Quantum computation promises to solve many hard or infeasible problems substantially faster than classical solutions. The involvement of big players like Google, IBM, Intel, Rigetti, or Microsoft furthermore led to a momentum which increases the demand for automated design methods for quantum computations. In this context, decision diagrams for quantum computation provide a major pillar as they allow to efficiently represent quantum states and quantum operations which, otherwise, have to be described in terms of exponentially large state vectors and unitary matrices. However, current decision diagrams for the quantum domain suffer from a trade-off between accuracy and compactness, since (1) small errors that are inevitably introduced by the limited precision of floating-point arithmetic can harm the compactness (i.e., the size of the decision diagram) significantly and (2) overcompensating these errors (to increase compactness) may lead to an information loss and introduces numerical instabilities.**

In this work, we describe and evaluate the effects of this trade-off which clearly motivates the need for a solution that is perfectly accurate *and* compact at the same time. More precisely, we show that the trade-off indeed weakens current design automation approaches for quantum computation (possibly leading to corrupted results or infeasible run-times). To overcome this, we propose an alternative approach that utilizes an algebraic representation of the occurring complex and irrational numbers and outline how this can be incorporated in a decision diagram which is suited for quantum computation. Evaluations show that—at the cost of an overhead which is moderate in many cases—the proposed algebraic solution indeed overcomes the trade-off between accuracy and compactness that is present in current numerical solutions.

***Index Terms*—Quantum Computing, Decision Diagrams, Algebraic Number Representation, Clifford+T**

I. Introduction

Quantum computation [1] received significant attention over the recent years since it constitutes a complementary computation paradigm that promises substantial speed-ups for many hard or infeasible problems (in comparison to the best-known classical algorithms running on conventional computers). While the computational entities of a conventional computer can be in either of the two basis states 0 and 1 only, the qubits of a quantum computer can assume an (almost) arbitrary superposition of both basis states. This superposition serves as basis for the so-called quantum parallelism, which, in combination with other quantum mechanical effects like entanglement and phase shifts, allows for substantial speed-ups in many applications.

But even though the basic idea of quantum computation as well as corresponding algorithms with remarkable speed-ups are around for several decades [2], [3], physical realizations are still in their infancy. However, big players like Google, IBM, Intel, Rigetti, or Microsoft heavily invest into research on quantum computers—carrying out a race for the first useful quantum computer [4]. This led to a new momentum in this domain with frequent "breakthroughs", e.g. in increasing the number of available qubits and their rapidly improving fidelity. Hence, in order to be prepared for future quantum devices, also research on automated design methods for quantum computations is underway (see e.g. [5], [6] for methods targeting the technology mapping of quantum circuits to actual quantum computers and [7] for a study on the detection and diagnosis of faulty quantum gates). These methods of course rely on representations for the corresponding states and operations. Since quantum computations are usually described in terms of exponentially large state vectors and unitary matrices, this often leads to rather intractable solutions when using straightforward representations like, e.g., 1- and 2-dimensional arrays [8]–[10].

Motivated by that, alternative representations are currently investigated. Inspired by the conventional domain—where design tasks often utilize compact representations such as *Binary Decision Diagrams* (BDDs [11])—decision diagrams are considered a promising approach for the efficient representation of quantum computations as well [12]–[15]. In fact, in many practically relevant cases the considered functionality exhibits redundancies which allow for more compact, non-exponential representations when employing decision diagrams. Since there also exist algorithms to efficiently manipulate these representations (with polynomial complexity with respect to the size of the decision diagrams), this allows to conduct certain design tasks in an efficient fashion. In this regard, especially the *Quantum Multiple-valued Decision Diagram* (QMDD [15]) is a promising representative and is actively being investigated—leading to a variety of efficient approaches e.g. for synthesis [16]–[19], verification [20]–[23], and simulation [24], [25].

Philipp Niemann and Rolf Drechsler are with the Group for Computer Architecture, University of Bremen, D-28359 Bremen, Germany, and the Cyber-Physical Systems Department, DFKI GmbH, D-28359 Bremen, Germany, E-mail: pniemann@uni-bremen.de / drechsler@uni-bremen.de

Alwin Zulehner and Robert Wille are with the Institute for Integrated Circuits, Johannes Kepler University Linz, A-4040 Linz, Austria, E-mail: alwin.zulehner@jku.at / robert.wille@jku.at

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

However, current decision diagrams for the quantum domain suffer from a trade-off between accuracy and compactness:[1]

- On the one hand, small errors are inevitably introduced by the limited precision of floating-point arithmetic on conventional computers. In fact, the complex numbers in the state vectors and transformation matrices often have *irrational* imaginary or real parts that can only be approximated. If these possible errors are not taken into account, redundancies might not be recognized—which can harm the compactness (i.e. the size of the decision diagram) significantly. That is, a small amount of inaccuracies has to be tolerated in order to find redundancies and, thus, achieve compactness.
- If, on the other hand, too much inaccuracy is tolerated, this leads to a loss of information and introduces numerical instabilities that—in the worst case—will falsify the results. While for some applications a moderate error may be acceptable (e.g., since the underlying quantum algorithm is robust enough), others require a rather accurate representation.

Consequently, an application-specific trade-off between accuracy and compactness needs to be conducted thus far in order to obtain efficient and sufficiently accurate methods on a case-by-case basis (such as for those discussed in [17], [19], [20], [23], [24]). Even more, a time-consuming fine-tuning of the corresponding parameters can be necessary in order to adapt design methods to a certain functionality or algorithm. However, as confirmed by the first thorough analysis of this issue that will be conducted in this work, it is not guaranteed that the desired accuracy or compactness can be achieved at all which—in the worst case—might cause corrupted results or infeasible run-times, respectively. These observations clearly support the need for an alternative solution that allows to overcome the trade-off present in current solutions and inherently achieves accuracy *and* compactness at the same time.

In this work[2], we are addressing this need with the following contributions:

- We propose such an alternative approach in which the considered quantum functionality is represented algebraically rather than numerically. By this, the proposed decision diagram can fully exploit existing redundancies for a compact representation and, at the same time, guarantees a perfectly accurate result—thereby completely avoiding the trade-off between accuracy and compactness.
- This solution allows to evaluate and quantify the existing trade-off between accuracy and compactness in decision diagrams for quantum computation. More precisely, we show how the current trade-off between accuracy and compaction indeed weakens current design automation approaches for quantum computation (leading to completely wrong results or infeasible run-times).
- The overhead of the proposed solution compared to current solutions is moderate in many cases although it guarantees perfect accuracy (which cannot be reached by numerical approaches due to machine accuracy) and compactness at the same time.

The remainder of the paper is structured as follows: the following Section reviews the basics of quantum computation as well as corresponding decision diagrams. To this end, a particular focus is put on *Quantum Multiple-valued Decision Diagrams*. Afterwards, the trade-off between accuracy and compactness that emerges when using decision diagrams in the quantum domain is discussed in Section III. Section IV then describes in detail how this issue can be addressed by using an algebraic number representation and how this representation can be exploited in decision diagrams in order to achieve both, perfect accuracy *and* compactness. In Section V, we evaluate and quantify the trade-off between accuracy and compactness that is present in current solutions, as well as the overhead required to overcome this issue by using the proposed algebraic representation. Section VI concludes the paper.

II. Background

This section briefly reviews the basics of quantum computation. Furthermore, we introduce the basic ideas of *Quantum Multiple-valued Decision Diagrams (QMDDs)*, a data-structure used to efficiently represent quantum functionality and which the proposed approach is built on. For a more detailed introduction, we refer to [1] and [15], respectively.

A. Quantum Computation

The computational entities of a quantum system are called *qubits*. A qubit has two *basis states* (usually denoted as $|0\rangle$ and $|1\rangle$) which can be seen as the analogue of the two possible states of a conventional bit. However, according to the Dirac-von Neumann formalization of quantum mechanics, the state of a qubit can be any linear combination $\alpha_0|0\rangle + \alpha_1|1\rangle$ for complex-valued α_0, α_1 with $|\alpha_0|^2 + |\alpha_1|^2 = 1$, i.e. any *superposition* of the basis states. Accordingly, an n-qubit quantum system can be in one of 2^n basis states ($|0\dots00\rangle$, $|0\dots01\rangle, \dots, |1\dots11\rangle$) or a superposition of these states. The state of such a quantum system is represented by a *state vector* of dimension 2^n where the i-th component α_i is called the *amplitude* of basis state $|i\rangle$. In general, it is not possible to completely determine the state of a physical quantum system, i.e. all amplitudes of the state vector. Instead, measuring the quantum system will let it collapse to some basis state (with non-zero amplitude) where the probability of measuring a particular basis state $|i\rangle$ is given by $|\alpha_i|^2$.

Example 1. *The basis states of a qubit are given as* $|0\rangle = \binom{1}{0}$ *and* $|1\rangle = \binom{0}{1}$*, while* $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\binom{1}{1}$ *and* $|-\rangle = \frac{1}{\sqrt{2}}\binom{1}{-1}$ *denote states of balanced superposition, i.e. where both basis states are equally likely to being measured.*

In order to use quantum systems for computation, the state of a quantum system can be modified by applying quantum operations. These are described by a $2^n \times 2^n$ *unitary transformation matrix*, i.e. an invertible complex-valued matrix whose inverse is given by the adjoint matrix.

[1] Note that we are considering QMDDs in the following. However, the problem discussed here as well as the proposed solutions are also applicable to other types of decision diagrams for quantum computation like [12], [14].

[2] A preliminary version of this work is available at [26].

Example 2. *In order to set a qubit into balanced superposition (e.g., $|+\rangle$ or $|-\rangle$), the* Hadamard *operation H is employed whose transformation matrix is given as $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. In fact, the transformed quantum state can be computed via matrix-vector multiplication as follows:*

$$H \cdot |0\rangle = H \cdot \begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle.$$

Further, frequently used quantum operations include the NOT *operation X (flipping the basis states $|0\rangle$ and $|1\rangle$) as well as the* phase shift *operations T ($\pi/4$ gate), $S = T^2$ (Phase gate) and $Z = S^2$. The corresponding unitary matrices are defined as*

$$X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix},\ T = \begin{pmatrix}1 & 0\\ 0 & \omega\end{pmatrix},\ S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix},\ Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix},$$

where $\omega = \frac{1+i}{\sqrt{2}} = e^{i\pi/4}$. Besides these operations that are applied to a single target qubit*, there are also controlled operations on multiple qubits. The state of the additional* control qubits *determines which operation is performed on the target qubit. An example is the* controlled NOT *(CNOT) operation on two qubits whose transformation matrix is defined by*

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}.$$

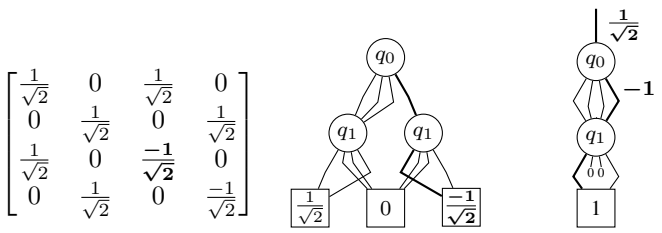
This operation performs a NOT operation on the target qubit if, and only if, the control is in the $|1\rangle$-state. All remaining qubits are not affected by the operation.

Complex, high-level quantum computations or algorithms need to be decomposed into a sequence of elementary quantum operations like the ones in Example 2 (so-called *quantum gates*) in order to execute them on a quantum computing device. The unitary matrix of a gate matrix is constructed as the Kronecker product of the base transformation matrix (e.g. H,T, or $CNOT$) together with identity matrices for all qubits that are neither target nor control for the gate. The unitary matrix of the entire high-level operation can then be computed as the matrix product of the individual gate matrices (in reversed order).

B. Decision Diagrams for Quantum Computation

Straight-forward representations of state vectors and unitary matrices like 1- or 2-dimensional arrays (e.g. those proposed in [8]–[10]) quickly become infeasible in the design of quantum logic. In fact, they often lead to exorbitant run-times or memory explosion already when being applied to quantum systems consisting of roughly 15-20 qubits. This is due to the exponential growth of the vectors/matrices with respect to the number of qubits.

Similar problems occur with truth-table representations of Boolean functions in conventional logic design and have been successfully addressed by using dedicated data-structures like *Binary Decision Diagrams* (BDDs, [11]), *Kronecker Functional Decision Diagrams* (KFDDs, [27]), or *Binary Moment Diagrams* (BMDs, [28]). These exploit redundancies by employing functional decompositions like, e.g., Shannon or Davio decompositions, in order to allow for a more compact representation and efficient manipulation of the considered functionality.



(a) Unitary matrix (b) DD w/o edge weights (c) QMDD

Fig. 1: Representations for $U = H \otimes I_2$.

As a consequence, several proposals for the use of dedicated data-structures in the quantum domain exist, e.g. *Quantum Decision Diagrams* (QDDs, [13]), *Quantum Information Decision Diagrams* (QuIDDs, [12]), *X-decomposition Quantum Decision Diagrams* (XQDDs, [14]) and *Quantum Multiple-Valued Decision Diagrams* (QMDDs, [15]).

The general idea of these approaches is to represent a (unitary) matrix or (state) vector in terms of a directed acyclic graph such that sub-matrices which occur multiple times are represented by a shared graph structure. In this regard, QMDDs have the unique property that they additionally make use of weighted edges. This allows them to use shared structures also for sub-matrices that differ by a scalar factor—a case that occurs frequently for the unitary matrices considered in quantum computation.

Example 3. *Figure 1a shows the transformation matrix of the quantum operation $U = H \otimes I_2$, i.e. a Hadamard operation is performed on one qubit of a 2-qubit quantum system. A decision diagram representation of this matrix is shown in Fig. 1b. Here, the single* root node *(labeled q_0) represents the whole matrix and has four outgoing edges to nodes representing the top-left, top-right, bottom-left, and bottom-right sub-matrix (from left to right). Likewise the 2×2 sub-matrices (represented by nodes labeled q_1) are decomposed until the terminal nodes are reached—each of which represents a distinct complex number.*

Apparently, the top-left, top-right and bottom-left sub-matrices of the original matrix are identical and can be represented by a shared graph structure (the left-most node labeled q_1 in Fig. 1b). However, the bottom-right sub-matrix is represented by a separate graph structure, although it has the same structure and differs only by a scalar factor of -1. If this similarity is taken into account (as it is done in QMDDs), an even more compact representation can be achieved. In fact, by extracting such scalar factors and annotating them to the corresponding edges, a single node at the q_1 level is sufficient as shown in the QMDD representation depicted in Fig. 1c. Here, the common factor $\frac{1}{\sqrt{2}}$ is extracted and annotated to the root edge *(an additional edge that points to the root node, but has no source). For simplicity, edge weights equal to 1 are suppressed and edges with weight 0 are indicated by stubs.*

To obtain the value of a particular matrix entry, one has to follow the corresponding path from the root to the terminal node and multiply all edge weights on this path. For example, the matrix entry $\frac{-1}{\sqrt{2}}$ from the bottom-left 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. 1c.

In order to determine which sub-matrices only differ by scalar factors and can, thus, make use of the enhanced compactness, the nodes of a QMDD are *normalized*. This means that a normalization factor is factored out from all outgoing edge weights and propagated to all incoming edges. By this, it is ensured that the overall products on all paths stay the same. There is some degree of freedom how the normalization factor is determined: for simplicity, one often takes the left-most non-zero weight of the outgoing edges, such that a node is normalized if, and only if, the left-most non-zero weight is 1. As an alternative, also the (left-most) edge weights with the largest absolute values can be used as normalization factors [29]. This ensures that all occurring edge weights will have an absolute value less than or equal to 1 which can increase the numerical stability of the representation—at the cost of a small computation overhead. In both cases, QMDD even become canonical, i.e. unique, representations of (unitary) matrices [15] which is an essential requirement for several design tasks. Indeed, QMDDs have shown to provide a compact representation of many practically relevant quantum functions and, thus, allow for an efficient processing of the respective matrices and vectors which led to powerful solutions for design tasks like synthesis [16]–[19], verification [20]–[23], and simulation [24], [25].

III. Accuracy vs. Compactness in QMDDs

The compression of QMDDs and other decision diagrams for quantum or conventional logic is a lossless one. This means that compactness is achieved by exploiting redundancies such that the entire information of the matrix, vector, or Boolean function, respectively, is preserved, i.e. it can—in principle—be reconstructed completely from the corresponding (QMDD) representation.

In the classical domain, gaining a compact representation without information loss is not complicated, neither from a mathematical nor an implementation point of view, since the set of possible values is finite (e.g. 0 and 1 for Boolean functions) or discrete (e.g. only integer numbers occur). As a consequence, decision diagrams that represent conventional computations like e.g. *Binary Decision Diagrams* (BDDs, [11], [30]–[32]), *Kronecker Functional Decision Diagrams* (KFDDs, [27]), or *Binary Moment Diagrams* (BMDs, [28]) do not face problems with the accuracy of the representation, since all values can be and are represented as (tuples) of integers—a strong canonical/unique form of representation.

This is different for the quantum domain, where we have to deal with arbitrary complex numbers. From a mathematical perspective, this does not causes problems since complex numbers also provide a strong canonical form. However, it introduces severe challenges from an implementation perspective where machine accuracy is limited and, hence, complex numbers (especially those with irrational coefficients) are approximated—yielding to numerical errors in computations and making accuracy an important issue for decision diagrams representing quantum computations.

To this end, first note that, in the area of quantum computation, most design automation tasks require hundreds or even thousands of matrix-matrix multiplications (e.g. to compute the unitary matrix for an entire quantum circuit from the gate matrices) or matrix-vector multiplications (e.g. to simulate the evolution of a quantum state during a quantum algorithm). These tasks do not constitute an issue per se, since the multiplication with a unitary matrix is a well-conditioned problem from a numerical perspective. In fact, the error in the result, i.e. the deviation from the exact result, can be expected to be in the order of the input error.[3] Furthermore, applying several multiplications successively will only lead to an error that grows linearly with the number of matrix multiplications. Consequently, using a numerical, i.e. approximated, representation of the complex numbers with a high resolution can yield numerically stable computations.

However, this approximation can have a significant impact on the decision diagram representation. To this end, recall that the key idea of decision diagrams is to exploit redundancies in order to gain a compact representation. This compact representation is indeed a key factor for their efficiency, since the complexity of the manipulation algorithms (e.g. matrix multiplication) grows with the size of the decision diagram.

However, this assumes that redundancies can be detected. While this is rather simple in the conventional domain, the occurring irrational numbers of the real and imaginary part of the complex numbers in quantum computing constitute a tough challenge. An example demonstrates the problem.

Example 4. *Recall Example 3 where the matrix shown in Fig. 1a can compactly be represented by the QMDD shown in Fig. 1c. This compact representation is possible since several redundancies can be exploited. However, representing the irrational entries with floating point numbers on a machine with limited accuracy, may break these redundancies e.g. when using rounding towards ∞ or when the matrix is constructed as the product of several other matrices. Then, two occurrences of $\pm\frac{1}{\sqrt{2}}$ might be represented by slightly different floating point numbers (differing in a few of the least significant bits of the mantissa) and no redundancy can be detected anymore.*

In general, this will likely lead to a matrix or a vector where no redundancies are detected at all—leading to an exponentially large representation. A solution to this issue that some of the redundancies that actually exist in the matrix are not detected due to tiny errors caused by the machine accuracy, is to identify numbers that do not differ by more than a so-called *tolerance value* (denoted as ϵ in the following). This approach has also been taken for QMDDs.

[3] Note that this is a statement about the matrix multiplication problem itself and not about a certain implementation.

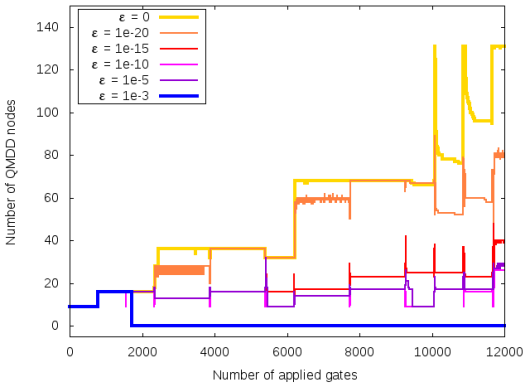


Fig. 2: Size of the QMDD when simulating GSE

Example 4 (continued)**.** *Assume that two entries that shall represent* $\frac{1}{\sqrt{2}}$ *differ only in the last three bits of the mantissa (assuming an IEEE 754 single precision floating point number with 23 mantissa bits). Then, setting e.g.* $\epsilon = 10^{-5}$ *allows to detect that the two entries are equal.*

However, choosing a proper value for ϵ is crucial. If ϵ is chosen too small, it might not be able to compensate the limited machine accuracy and, thus, to determine more redundancies. If ϵ is chosen too large, this might lead to numerical instabilities of the multiplication algorithm. Moreover, additional redundancies might be detected that are not actually present—leading to an undesired approximation and, thus, information loss. In the worst case, this may falsify the result such that an invalid quantum state (e.g. a vector composed of zeros only) or a non-unitary matrix results. Nevertheless, in many cases there exist proper configurations for ϵ, but this heavily depends on the considered application and determining an adequate tolerance value may require time-consuming fine-tuning of parameters on a case-by-case basis.

Example 5. *Fig. 2 shows the size of the QMDD through simulating the* Ground State Estimation *(GSE, [33]) quantum algorithm. This algorithm originates from quantum physics and computes the ground state energy of a quantum molecular system. As can be seen, the number of QMDD nodes is highly affected by* ϵ*. Choosing* $\epsilon = 0$*, i.e. (almost) no two different numbers are considered to be equal, yields the highest precision that is possible using floating point numbers, but results in a rather large representation. Instead, choosing* $\epsilon = 10^{-3}$ *yields a vector composed of zeros only—a perfectly compact but obviously wrong representation. Of course, both choices (highlighted in bold in Fig. 2) represent extreme cases. As a trade-off, choosing* $\epsilon = 10^{-15}$ *leads to almost the same numerical result as* $\epsilon = 0$*, but yields a better compactness and, thus, a smaller run-time.*

Overall, determining a "perfect" ϵ, i.e. finding the best trade-off between accuracy and compactness (which heavily influences the run-time), is a non-trivial task. So far, it has to be evaluated on a case by case level for each application.

In this work, we propose to overcome this trade-off by using an algebraic representation of the complex numbers that occur in the vectors/matrices—eventually resulting in a decision diagram that detects all existing redundancies and computes the result in an exact fashion (i.e. without a numerical error).

IV. Proposed Solution

In this section, we propose a solution for the algebraic representation of complex numbers in QMDDs which overcomes the approximation drawbacks caused by the numerical number representation and allows for both, a perfect accuracy together with a perfect exploitation of redundancies. To this end, we first discuss the properties of the ring $\mathbb{D}[\omega]$ that will be utilized for the exact, algebraic representation of complex numbers. After that, we present a solution for exploiting the benefits of this representation in QMDDs.

A. Utilizing the Ring $\mathbb{D}[\omega]$

In order to obtain an algebraic representation of the complex numbers, the most obvious choice would be to extend the well-known Gaussian numbers $\mathbb{Z}[i]$ to the ring $\mathbb{Z}[i, \sqrt{2}]$. By doing so, all complex numbers of the form $a + b\sqrt{2} + i(c + d\sqrt{2})$ can be represented exactly. This ring is already a dense subset of the complex numbers such that any complex number can be approximated by an element from $\mathbb{Z}[i, \sqrt{2}]$ up to an arbitrary precision (this *density* is a known property of $\mathbb{Z}[\sqrt{2}]$ in the real numbers and can easily be lifted to the complex numbers). However, the irrational number $\frac{1}{\sqrt{2}}$ that plays a vital role in quantum computation, is not contained in this ring.[4] Thus, it seems more promising to study the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ which trivially contains $\mathbb{Z}[i, \sqrt{2}]$ (since $\sqrt{2} = 2 \cdot \frac{1}{\sqrt{2}}$), but allows to represent $\frac{1}{\sqrt{2}}$ and all its potencies exactly.

In the following, we will make use of a different interpretation of this ring that is more convenient from an algebraic perspective. More precisely, we will use the interpretation as an extension of the so-called *dyadic fractions* $\mathbb{D} = \{\frac{a}{2^k} \mid a, k \in \mathbb{Z}, k \geq 0\}$, namely $\mathbb{D}[\omega]$ for the complex number $\omega = \frac{1+i}{\sqrt{2}} = e^{i\pi/4}$ (as in Example 2).[5]

Using the latter representation, all complex numbers that can be represented exactly can be written as

$$\alpha = \frac{1}{\sqrt{2}^k}(a\omega^3 + b\omega^2 + c\omega + d)$$

for coefficients $a, b, c, d, k \in \mathbb{Z}$, i.e. using five integers (cf. [8]).

Example 6. *The irrational number* $\sqrt{2}$ *can be represented as* $\frac{1}{\sqrt{2}^{-1}}(0\omega^3 + 0\omega^2 + 0\omega + 1)$*, i.e., with* $k = -1$*. Besides that, also representations with* $k = 0$ *or* $k = 1$ *are possible, i.e.,* $\frac{1}{\sqrt{2}^0}(-\omega^3 + 0\omega^2 + \omega + 0) = \omega - \omega^3 = \frac{1+i}{\sqrt{2}} - \frac{-1+i}{\sqrt{2}} = \frac{2}{\sqrt{2}} = \frac{1}{\sqrt{2}^1}(0\omega^3 + 0\omega^2 + 0\omega + 2) = \sqrt{2}$.

Note that the ring $\mathbb{D}[\omega]$ is also strongly related to the well established Clifford+T gate library [34]. This library is very

[4] If it was, then also $\frac{1}{2}$ would be a member and could be written as $\frac{1}{2} = a' + b'\sqrt{2}$ for some $a', b' \in \mathbb{Z}$. However, since it must hold that $b' \neq 0$, this immediately yields the contradiction $\sqrt{2} = \frac{1-2a'}{2b'} \in \mathbb{Q}$.

[5] The fact that the rings $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ and $\mathbb{D}[\omega]$ are isomorphic becomes obvious if one considers the ring $\mathbb{D}[\sqrt{2}, i]$ (which can easily be seen to be isomorphic to both rings) as an intermediate step. In fact, $\sqrt{2} = \omega - \omega^3$ and $i = \omega^2$.

popular in quantum computation due to its *universality* (any quantum operation, i.e. any unitary transformation matrix, can be realized up to an arbitrarily small error) as well as *fault-tolerance* (robust, fault-tolerant implementations of these gates are known for most technologies that are considered promising for large-scale quantum computers). The most elementary gates in this library are the Clifford group gates (H, CNOT, S) and the T gate as discussed in Example 2. The relation between the ring $\mathbb{D}[\omega]$ and the Clifford+T gate library is that the quantum operations which can be realized exactly by Clifford+T gates (i.e. without any rounding error) are precisely given by those matrices whose entries are from the ring $\mathbb{D}[\omega] = \mathbb{D}[\sqrt{2}, i]$ (as shown in [8]). As a consequence, all such quantum operations can be represented with perfect accuracy using our approach. Furthermore, any quantum state and operation can be approximated to an arbitrary precision, since $\mathbb{D}[\omega]$ is a dense subset of the complex numbers. Hence, $\mathbb{D}[\omega]$ provides the ideal basis for a decision diagram that employs an accurate, algebraic representation of complex numbers.

B. Incorporating $\mathbb{D}[\omega]$ into QMDDs

In order to use the algebraic representation of complex numbers presented above within QMDDs, there are two aspects that have to be taken into account:

1) In order to determine common factors and structural similarities (that are required to find redundancies), a unique representation of $\mathbb{D}[\omega]$ numbers is required. However, there are in general infinitely many possibilities to represent a $\mathbb{D}[\omega]$ number (c.f. Example 6).
2) The extracted normalization factors have to be applied to the edge weights (c.f. Section II-B). More precisely, the weights have to be divided by these factors. However, as division means multiplication by the (multiplicative) inverse, this division can only be conducted properly for $\mathbb{D}[\omega]$ numbers that indeed have a multiplicative inverse in $\mathbb{D}[\omega]$, but not for $\mathbb{D}[\omega]$ numbers in general (e.g. all odd integers greater than or equal to 3 do not have an inverse in $\mathbb{D}[\omega]$ and the result of a division by such a number can not be represented as a $\mathbb{D}[\omega]$ number).

We propose to address these issues as follows:

1) Recall that each number from $\mathbb{D}[\omega]$ can be written as

$$\alpha = \frac{1}{\sqrt{2}^k}(a\omega^3 + b\omega^2 + c\omega + d)$$

for coefficients $a, b, c, d, k \in \mathbb{Z}$. If the exponent k is fixed, the representation is clearly unique since two different representations would yield a non-trivial representation of 0 in $\mathbb{Z}[\omega]$.[6] Thus, a unique representation can be achieved when using the *smallest denominator exponent* k_{min} such that there is no representation with an exponent $k < k_{min}$.

[6]This would contradict the fact that the potencies $\omega^0 = 1, \omega, \omega^2, \omega^3$ are linearly independent over $\mathbb{Z}$ (even over $\mathbb{Q}$), since ω is a primitive 8-th root of unity and the cyclotomic field $\mathbb{Q}[\omega]$ is a 3-dimensional vector space over $\mathbb{Q}$.

Algorithm 1: Compute Minimal $\mathbb{D}[\omega]$ Representation

Data: $\alpha = \frac{1}{\sqrt{2}^k}(a\omega^3 + b\omega^2 + c\omega + d) \in \mathbb{D}[\omega] \setminus 0$
Result: Representation of α with smallest denominator exponent $k_{\min}$

$a' \leftarrow a, b' \leftarrow b, c' \leftarrow c, d' \leftarrow d, k' \leftarrow k$
while $a' = c' \mod 2$ *and* $b' = d' \mod 2$ **do**
  $a' \leftarrow b' - d'$
  $b' \leftarrow c' + a'$
  $c' \leftarrow b' + d'$
  $d' \leftarrow c' - a'$
  $k' \leftarrow k' - 1$
// Criterion for minimality is satisfied
$k_{\min} \leftarrow k'$
return $\alpha = \frac{1}{\sqrt{2}^{k_{\min}}}(a'\omega^3 + b'\omega^2 + c'\omega + d') \in \mathbb{D}[\omega]$

The existence of such an exponent has already been discussed in [8], but no constructive criterion for minimality has been derived. To this end, we note that $\sqrt{2} = -\omega^3 + \omega$, such that

$$\begin{aligned}\alpha &= \frac{1}{\sqrt{2}^k}(a\omega^3 + b\omega^2 + c\omega + d) \cdot \frac{\sqrt{2}}{\sqrt{2}} \\ &= \frac{(b-d)\omega^3 + (c+a)\omega^2 + (b+d)\omega + (c-a)}{\sqrt{2}^{k+1}} \\ &= \frac{1}{\sqrt{2}^{k-1}}\left(a'\omega^3 + b'\omega^2 + c'\omega + d'\right)\end{aligned}$$

where $a', b', c', d' \in \mathbb{Z}$ if, and only if, $a = c \mod 2$ and $b = d \mod 2$. Thus, we know that the exponent is minimal if, and only if, $a \neq c \mod 2$ or $b \neq d \mod 2$.[7]

Example 7. *As already discussed in Example 6, the number $\sqrt{2}$ can be represented with $k = 0$ as $\frac{1}{\sqrt{2}^0}(-\omega^3+0\omega^2+\omega+0)$, i.e. with $b = d = 0$ and $a = -c$. Thus, the criterion for the smallest denominator exponent is not satisfied here. However, since $0 \neq 1 \mod 2$ the criterion is satisfied for $\frac{1}{\sqrt{2}^{-1}}(0\omega^3 + 0\omega^2 + 0\omega + 1)$, such that $k_{min} = -1$ in this case.*

In summary, the above consideration yields a constructive algorithm to obtain unique representations of $\mathbb{D}[\omega]$ numbers that is summarized in Algorithm 1.

2) Regarding the division by normalization factors, there are two viable alternatives. The first option is to employ the algebraic closure of $\mathbb{D}[\omega]$, namely $\mathbb{Q}[\omega]$. In this (cyclotomic) number field [35], a similar argumentation as above can be performed. In fact, each $\mathbb{Q}[\omega]$ number has a unique representation as $\frac{\alpha}{e}$ where $\alpha \in \mathbb{D}[\omega]$ and e is an odd integer ($e \in 2\mathbb{Z} + 1$) that is co-prime to the integer coefficients of α, i.e. $gcd(a, b, c, d, e) = 1$.
Having this, all computations can be made in the field $\mathbb{Q}[\omega]$ where all non-zero numbers have a multiplicative

[7]Apparently, there is no smallest denominator exponent for 0, such that we define the unique representation of 0 as $a = b = c = d = k = 0$.

Algorithm 2: Normalization with $\mathbb{Q}[\omega]$ Inverses

Data: QMDD node v with weights $w_{00}, w_{01}, w_{10}, w_{11}$ of outgoing edges (not all zero)
Result: Normalized node v' and normalization factor η

```
// Determine the leftmost non-zero
   edge weight
```
$i \leftarrow 0$
while $w_i = 0$ **do**
  $i \leftarrow i + 1$
```
// Divide all weights by this weight
```
$\eta \leftarrow w_i$
for $j \leftarrow i$ **to** 3 **do**
  $w_j \leftarrow w_j \cdot \overline{\eta} \cdot \frac{1}{N(\eta)}$
return *Node with updated weights,* η

inverse. This inverse can be constructed as follows: the squared norm $N(z)$ of a number $z \in \mathbb{Q}[\omega]$ is given as

$$N(z) = z \cdot \overline{z} = u + v\sqrt{2} \text{ for some } u, v \in \mathbb{Q}.$$

Using the third binomial formula, the inverse of $N(z)$ can hence be written as

$$\frac{1}{N(z)} = \frac{u - v\sqrt{2}}{u^2 - 2 \cdot v^2}.$$

Finally, the inverse of z is given by rewriting the first equation as

$$z^{-1} = \overline{z} \cdot \frac{1}{N(z)}.$$

Example 8. *Consider the number* $z = 1 + i\sqrt{2} \in \mathbb{D}[\omega]$. *The norm* $N(z)$ *is computed as* $(1 + i\sqrt{2}) \cdot (1 - i\sqrt{2}) = 1 - 2i^2 = 3$. *Thus,* $\frac{1}{N(z)} = \frac{3}{9} = \frac{1}{3}$ *and* $z^{-1} = \frac{1 - i\sqrt{2}}{3}$.

In summary, one possible way of performing normalization is spending one additional integer and switching to the algebraic number field $\mathbb{Q}[\omega]$. The corresponding normalization scheme is summarized in Algorithm 2.

The second option is to stay in $\mathbb{D}[\omega]$ and require that all normalization factors are common divisors of the edge weights. Then, the necessary divisions are clearly possible in $\mathbb{D}[\omega]$. As normalization shall achieve uniqueness, we suggest to take greatest common divisors (GCD) of the edge weights as normalization factors. Note that it is not a priori clear that GCDs indeed exist in $\mathbb{D}[\omega]$. However, we were able to show that $\mathbb{D}[\omega]$ is a Euclidean Ring which implies that GCDs exist and can be computed by iteratively applying the Euclidean algorithm. To this end, to show the existence of GCDs in $\mathbb{D}[\omega]$, we consider the sub-ring $\mathbb{Z}[\omega]$ with the function $E(z) = |(a^2 + b^2 + c^2 + d^2)^2 - 2 \cdot (ab + bc + cd + da)^2|$. One can show that this function is a Euclidean function such that for any non-zero $z_1, z_2 \in \mathbb{Z}[\omega]$ there exist $q, r \in \mathbb{Z}[\omega]$ with $z_1 = q \cdot z_2 + r$ and $E(r) \leq \frac{9}{16} E(z_2)$, i.e. $\mathbb{Z}[\omega]$ is a Euclidean ring and the Euclidean algorithm (that determines q by first performing the division z_1/z_2 in $\mathbb{Q}[\omega]$ and then rounding each component to the closest integer) will at some point terminate. As each $\mathbb{D}[\omega]$ number is *associated* to a $\mathbb{Z}[\omega]$ number, i.e. differs only by multiplication with a potency of the unit $\frac{1}{\sqrt{2}}$, this result can directly be extended to $\mathbb{D}[\omega]$.

As GCDs are only unique up to multiplication with units (i.e. invertible elements of the ring), we then apply a methodology to determine a GCD as normalization factor that yields that the leftmost non-zero edge weight z

a) is in $\mathbb{Z}[\omega]$, i.e. $k = 0$,
b) has a *minimal* norm, i.e. $N(z) = u + v\sqrt{2}$ such that one of the derived pairs $(|u|,|v|)$ and $(|2v|, |u|)$ is *minimal* among all associated $\mathbb{Z}[\omega]$ numbers w.r.t. lexicographical order (after factoring out potencies of 2), and
c) the quadruple of coefficients $(|a|, |b|, |c|, |d|)$ is the *lexicographical minimum* among all quadruples that can be derived by iteratively rotating $(|a|, |b|, |c|, |d|) \mapsto (|b|, |c|, |d|, |a|)$. Furthermore, d has positive sign.

Example 9. *The number* $\alpha = 2\omega^3 + 3\omega^2 + 2\omega + 4$ *from* $\mathbb{Z}[\omega]$ *satisfies properties a) and c). However, its norm* $N(\alpha) = 33 + 12\sqrt{2}$ *with derived pairs* $(33, 12)$ *and* $(24, 33)$ *is not minimal. In fact, the associated number* $\alpha' = \alpha \cdot (\omega - 1) = -2\omega^3 + \omega^2 - \omega - 6$ *has norm* $N(\alpha') = 42 - 9\sqrt{2}$ *with derived pairs* $2 \cdot (9, 21)$ *and* $(42, 9)$ *from which the first one is minimal. In order to also satisfy property c),* α' *is rotated to* $\omega^3 - \omega^2 - 6\omega + 2$.

In the following, we provide a brief sketch of why it is always possible to find a (unique) GCD that yields the desired properties of the edge weights. To this end, one can show that the group of units $\mathbb{D}[\omega]^*$ is generated by $\frac{1}{\sqrt{2}}$, ω, and $(\omega \pm 1)$. Since multiplying with $\frac{1}{\sqrt{2}}$ changes the norm of a number by $\frac{1}{2}$, we can restrict to numbers whose norm is in $\mathbb{Z}[\sqrt{2}]$, but not in $2\mathbb{Z}[\sqrt{2}]$. One can show, that each $\mathbb{Z}[\sqrt{2}]$ number $\eta = u + v\sqrt{2}$ has an *associated* number $\eta' = u' + v'\sqrt{2}$ that is reached by multiplication with a $\mathbb{Z}[\omega]$ unit, i.e. an appropriate potency of the $(1 \pm \sqrt{2})$ [36], where $|u'| < |v'|$ or $v = 0$ and this number is unique up to multiplication by -1. Having this in mind, we multiply the edge weight by $(\omega \pm 1)$ thereby multiplying its norm by $N(\omega \pm 1) = (2 \pm \sqrt{2}) = \pm\sqrt{2}(1 \pm \sqrt{2})$. By doing so, we can reach an edge weight whose norm is the associated number $\eta' = u' + v'\sqrt{2}$ from above (up to a potency of $\pm\sqrt{2}$, but this is not relevant, since we consider the pairs $(|u|,|v|)$ and $(|2v|, |u|)$ and factor out potencies of 2). In order to achieve the minimum of such pairs among all associated $\mathbb{Z}[\omega]$ numbers, we likewise compute $(\sqrt{2}N(z))'$ (again up to a potency of $\pm\sqrt{2}$) and accept that edge weight which exhibits the minimum among the derived pairs.

Finally, regarding the lexicographical order required on the coefficients of z, it is easy to see that the rotation corresponds to a multiplication by ω (which does not change the norm) and that a positive d can always achieved by multiplication with $-1 = \omega^4$. The existence of unique a lexicographical minimum is obvious in most

Algorithm 3: Normalization with GCDs from $\mathbb{D}[\omega]$

Data: QMDD node v with weights $w_{00}, w_{01}, w_{10}, w_{11}$ of outgoing edges (not all zero)
Result: Normalized node v' and normalization factor η

```
// Determine a GCD of all weights
g ← gcd(w_00, w_01, w_10, w_11)
// Determine the leftmost non-zero
   edge weight
i ← 0
while w_i = 0 do
    i ← i + 1
// Adjust GCD in order to achieve
   desired edge weight properties
z ← w_i / g
z' ← reduceNorm(z)
z' ← lexicographicalMinimum(z')
k' ← exponent k of z'
z' ← z' · √2^{k'}
g' = g · (z / z_i)
// Divide all weights by this GCD
η ← g'
for j ← i to 3 do
    w_j ← w_j / η
return Node with updated weights, η
```

cases, aside from the cases that 1) $|a| = |b| = |c| = |d|$ as well as 2) $|a| = |c|$ and $|b| = |d|$. However, in both cases the exponent k is not minimal according to the criterion derived above and a potency of $\frac{1}{\sqrt{2}}$ can be factored out until k_{min} is reached.

Thus, a unique normalization with normalization factors in $\mathbb{D}[\omega]$ is possible as summarized in Algorithm 3.

Overall, the proposed solution allows for the algebraic and, thus, perfectly accurate representation of complex numbers within the QMDD data-structure. However, this comes at the price of more expensive arithmetic operations and the resulting computational overhead will be evaluated in Section V-B.

V. Evaluations

In this section, we present the results of our experimental evaluations. More precisely, we conducted a detailed evaluation on the current trade-off between accuracy and compactness in decision diagrams for quantum computations following the state-of-the-art *numerical QMDD representation* (which utilizes floating point numbers in the IEEE 754 double precision format to represent irrational coefficients and supports the configuration of a tolerance value ϵ).

Note that this evaluation only becomes possible by having a perfectly accurate solution, namely the proposed *algebraic QMDD representation*, to compare with. In a second step, we evaluated how the proposed solution overcomes this trade-off. Note that we used the GNU Multiple Precision Arithmetic Library (GMP, [37]) for representing the integers occurring in the algebraic number representations.

Since no established benchmark suite for quantum computation is available yet, we considered a self-compiled benchmark suite consisting of well-known quantum algorithms covering different classes and application domains of quantum algorithms. More precisely, Grover's algorithm [2] and the *Binary Welded Tree* algorithm (BWT, [38]) address problems from the computer science domain (database search and graph exploration) and all quantum gates and complex numbers occurring during the computation are exactly representable by the proposed algebraic approach, i.e. without any approximation error. In contrast, the GSE algorithm introduced in Example 5 addresses a problem from quantum physics (estimation of the ground state energy of a quantum molecular system). It represents a class of quantum algorithms whose original description is not directly compatible with the proposed solution, since the required quantum operations (e.g. rotations by arbitrary angles) result in complex numbers that are not contained in $\mathbb{D}[\omega]$ or $\mathbb{Q}[\omega]$, respectively, and need to be approximated.

For this purpose, we extracted suitable approximations in terms of quantum circuits consisting solely of (exactly representable) Clifford+T gates using the Quipper tool [39].

All benchmarks have then be simulated by representing the corresponding quantum states and quantum operations using the QMDD package taken from [40] on a classical 3.8 GHz machine with 32 GB of memory. Note that an additional simulation on a physical quantum computer was not possible since the considered benchmarks by far outreach what is computable on currently available state-of-the-art quantum computers (so called NISQ devices) concerning the number of qubits considered, the number of quantum operations performed, and the accuracy of the obtained results.

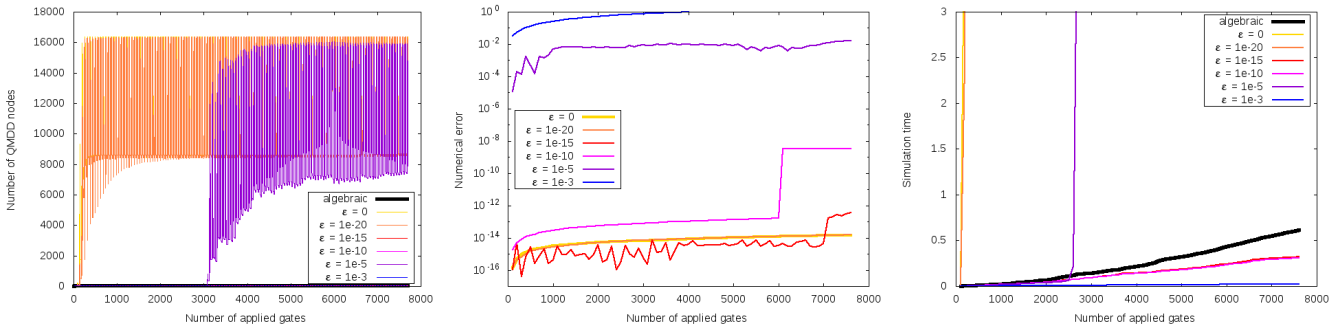
In the following, a summary of the respectively obtained results is provided and discussed.

A. Trade-off Between Accuracy and Compactness

In a first series of evaluations, we investigated the accuracy and the compactness (as well as the impact on the simulation run-time) of the recently applied, i.e. *numerical*, QMDD representation for different values of ϵ. In the following, we discuss the obtained results for Grover's algorithm as well as for the BWT and GSE algorithm whose results are provided in Fig. 3, Fig. 4, and Fig. 5, respectively, and which provide good representatives of our evaluations. For each quantum algorithm, we provide graphs showing the size (i.e. the number of nodes) of the QMDD that represents the evolved quantum state, the accuracy throughout the simulation[8], as well as the run-time of the simulation (in CPU seconds).

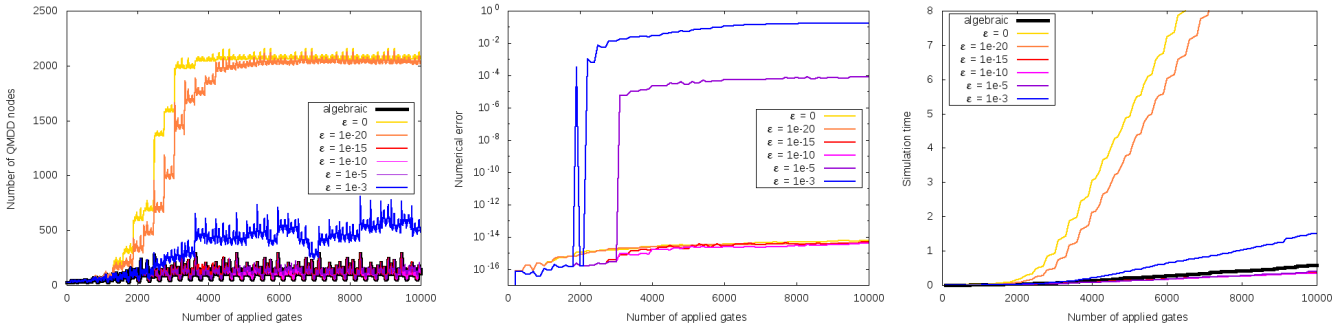
First of all, the results clearly confirm the general numerical stability of the QMDD-based matrix multiplication (cf. Section III). In fact, for a sufficiently small tolerance value ϵ, the error indeed scales linearly with the number of applied gates.

[8]In order to quantify the accuracy, we determine the relative deviation of the vector resulting from the numerical computation v_{num} from the algebraic (and, thus, exact) result v_{alg}. More precisely, we compute the Euclidean norm of $v_{num} - v_{alg}$ to quantify the loss of precision relative to the exact result. To have a fair evaluation, we adjust the norm of the numerically computed vector v_{num} to 1, since an error in the length of the vector can be fixed easily (except for a 0-vector).



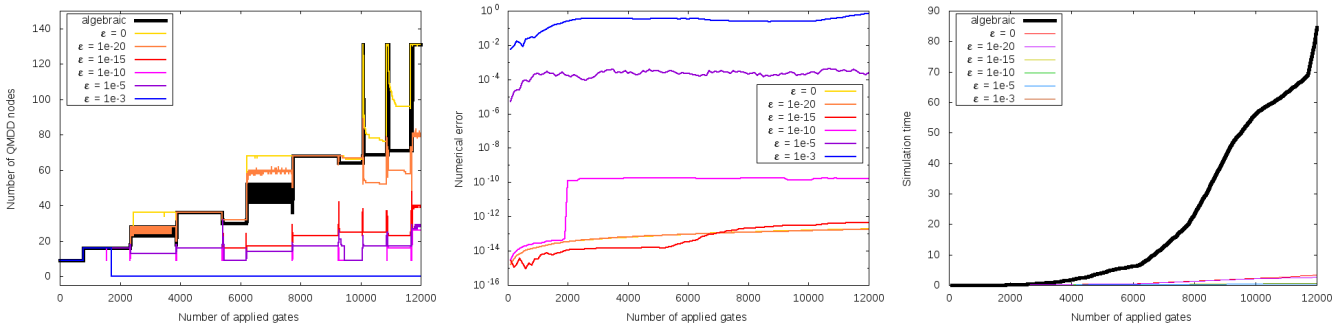
(a) Compactness (b) Accuracy (c) Simulation time

Fig. 3: Results for simulating Grover's algorithm



(a) Compactness (b) Accuracy (c) Simulation time

Fig. 4: Results for simulating the BWT algorithm



(a) Compactness (b) Accuracy (c) Simulation time

Fig. 5: Results for simulating the GSE algorithm

In addition, the provided plots also show that the compactness of the QMDD directly correlates with the simulation time. More precisely, the slope of the simulation times depicted in Fig. 3c, Fig. 4c, and Fig. 5c is proportional to the respective number of QMDD nodes.

However, the results also clearly confirm the discussed trade-off between accuracy and compactness. Consider for example the plots obtained for simulating Grover's algorithm using 15 qubits (i.e. Fig. 3):

- Using a numeric QMDD representation with a high accuracy ($\epsilon = 0$ or $\epsilon = 10^{-20}$) hardly allows to detect any redundancy and, thus, requires exponentially many nodes and a significant run-time.
- In contrast, choosing a moderate accuracy ($\epsilon = 10^{-15}$ or $\epsilon = 10^{-10}$) allows to detect more of the actually present redundancies and, hence, yields a quite compact representation. On the downside, this truncation leads to numerical issues. For instance, while choosing $\epsilon = 10^{-15}$ yields a rather small numerical error, the peaks in the graph indicate an undesired numerical instability in the multiplication algorithm that may lead to severe rounding errors in certain simulations.
- By choosing a low accuracy ($\epsilon = 10^{-5}$ or 10^{-3}) the accuracy of the QMDDs drop significantly—resulting in completely useless simulation results. Surprisingly, the number of QMDD nodes even increases exponentially for $\epsilon = 10^{-5}$ after applying approximately 3000 gates. However, this is an exceptional case since in the vast majority of the cases, increasing ϵ indeed increases the compactness of the numeric QMDD (as also confirmed by the results for the other quantum algorithms, c.f. Fig. 4a and Fig. 5a). As an extreme case, even a dropping down to zero can be observed (e.g. when choosing $\epsilon = 10^{-3}$) which obviously is a completely wrong result.

Overall, the provided plots clearly show the correlation between accuracy and compactness. In fact, the compactness of the QMDD heavily depends on the chosen accuracy (in terms of ϵ). While increasing ϵ yields a more compact representation and, thus, reduces run-time, it increases the probability for obtaining severe numerical errors—resulting in completely useless results (e.g. a zero-vector) in the worst case. As shown above, a good choice of ϵ depends on the considered problem instance and it can be quite difficult to choose it such that exactly those redundancies are found that are actually present.

In addition, the plots also show that, even when using a tolerance value of $\epsilon = 0$, i.e. when employing the highest possible precision, there is a lower bound to the numerical error that is never underrun. Even when scaling up the precision/bitwidth of the floating-point numbers—an investment that will likely lead to substantial run-time degradations—the same effect can be expected. In other words, the limited precision of the floating-point arithmetic will never allow for perfect accuracy (on the long run).

B. Evaluation of the Algebraic Representation

The algebraic QMDD solution proposed in this work overcomes the limitations that have been observed in the previous evaluation. In fact, there is no more need for determining an adequate accuracy for the problem at hand on a case-by-case basis. The algebraic QMDD will always achieve the maximum compactness that is possible without losing information (i.e. only exploiting redundancies that are actually present). Moreover, it will achieve perfect accuracy also on the long run which can be very important for design tasks like verification. For instance, checking equivalence of two matrices or vectors then boils down to comparing the root nodes of the corresponding QMDDs (which can be done in $O(1)$) instead of looking for (tiny) deviations in the whole representations.

However, the algebraic representation of complex numbers requires to perform arithmetic operations in the ring $\mathbb{Q}[\omega]$. These can induce a computation overhead as the integer coefficients may (in theory) become arbitrary large, while floating point arithmetic can often benefit from existing hardware accelerators e.g. in terms of a dedicated floating point unit. In the following, we evaluate this computation overhead in detail.

To this end, again Grover's as well as the BWT and GSE quantum algorithm as discussed above provide a good representative of our evaluations (also including an example with a worst case overhead). Accordingly, we also generated corresponding algebraic QMDDs and report the respectively obtained QMDD sizes and run-times by means of the the bold black graphs in Figs. 3-5. [9]

As can be seen, for Grover's as well as for the BWT algorithm, the algebraic QMDDs remain quite compact. They, thus, perform much better than the numerical QMDDs with high accuracy ($\epsilon = 0$ and $\epsilon = 10^{-20}$) that can not take advantage of the present redundancies. In comparison to the numerical QMDDs that exploit these redundancies, the algebraic QMDDs have a reasonable constant run-time overhead (around a factor of 2).

In contrast, the GSE algorithm is a representative for those cases, where the behavior is quite different. As can be seen, there are hardly any redundancies that can be exploited such that the size of the algebraic QMDD stays in the range of the sizes of the high accuracy numerical QMDDs. However, unlike in the previous cases, the run-times of the algebraic QMDD do not stay in the range of those numerical QMDDs showing the same sizes. In fact, they do not stay even close to the run-times of *any* numerical QMDD, but the computation overhead grows significantly. A more detailed analysis shows that this can be explained by the fact that the bit-widths of the integers used for algebraically representing the occurring complex numbers grow significantly. A possible explanation for this behavior is that the Clifford+T approximation of the GSE algorithm leads to complex numbers that are very costly to be represented and processed in an exact, algebraic way, while the numeric QMDDs are rather insensitive to particular complex numbers.

[9] Note that no graph is provided for accuracy as the proposed algebraic representation always is exact, i.e. does not include errors.

Moreover, only the run-times for the first normalization scheme (using normalization factors in $\mathbb{Q}[\omega]$) are shown, since these always outperformed the normalization scheme that uses GCDs as normalization factors.

In the normalization scheme using $\mathbb{Q}[\omega]$ numbers, the growth of the bit-widths is most significant for the denominators, while in the normalization indeed achieves that at least half of the occurring edge weights are trivial (equal to 1). This greatly simplifies the arithmetic operations and is beneficial for the overall runtime. In contrast, the normalization based on GCDs in most cases only obtains trivial GCDs to be factored out. This results in a less beneficial factorization with very few trivial edge weights and many weights with large coefficients that lead to a higher performance degradation.

Overall, in several cases the structural benefits of algebraic QMDDs in comparison to numerical QMDDs really become effective as the computation overhead remains small, while in other cases we obverse a significant overhead which might be the showstopper for algebraic QMDDs in such cases.

VI. Conclusions

In this work, we thoroughly discussed and evaluated the trade-off between accuracy and compactness of decision diagrams for quantum computation. Since this requires fine-tuning of parameters on a case-to-case basis and might still yield useless results, we propose to overcome this issue by an algebraic decision diagram. The proposed algebraic representation guarantees perfect accuracy while remaining compact (all redundancies that are actually present are detected). Experimental evaluations confirm the trade-off in the numerical representations and shows that the overhead of the algebraic solution is moderate in many cases and actually has no effect on the scalability in general which is comparable to previously proposed DDs for quantum computing. However, differences in scalability may occur when precision is traded-off against compactness (as also evaluated in Section V.A), i.e., if a less accurate result is acceptable, better scalability can be achieved with numerical approaches. If instead best accuracy is demanded, scalability might be affected. This, however, is not a result of the overhead caused by the proposed algebraic representation (which, again, is moderate), but a consequence of the demand for exact results. Future work covers research on new normalization schemes for the algebraic representation in order to further reduce the overhead—particularly for cases where this contributes to a large overhead.

VII. Acknowledgments

This work was partially supported by the LIT Secure and Correct System Lab funded by the State of Upper Austria.

References

[1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symposium on the Theory of Computing*, 1996, pp. 212–219. [Online]. Available: http://doi.acm.org/10.1145/237814.237866

[3] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172

[4] L. Gomes, "Quantum computing: Both here and not here," *IEEE Spectrum April 2018*, 2018.

[5] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 7, pp. 1226–1236, 2019. [Online]. Available: https://doi.org/10.1109/TCAD.2018.2846658

[6] Y. Lin, B. Yu, M. Li, and D. Z. Pan, "Layout synthesis for topological quantum circuits with 1-D and 2-D architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1574–1587, 2018. [Online]. Available: https://doi.org/10.1109/TCAD.2017.2760511

[7] D. Bera, "Detection and diagnosis of single faults in quantum circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 3, pp. 587–600, 2018. [Online]. Available: https://doi.org/10.1109/TCAD.2017.2717783

[8] B. Giles and P. Selinger, "Exact synthesis of multiqubit Clifford+T circuits," *Phys. Rev. A*, vol. 87, no. 3, p. 032332, Mar. 2013.

[9] D. S. Steiger, T. Häner, and M. Troyer, "ProjectQ: an open source software framework for quantum computing," *arXiv preprint arXiv:1612.08091*, 2018.

[10] N. Khammassi, I. Ashraf, X. Fu, C. Almudever, and K. Bertels, "QX: A high-performance quantum computer simulation platform," in *Design, Automation and Test in Europe*, 2017.

[11] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[12] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Improving gate-level simulation of quantum circuits," *Quantum Information Processing*, vol. 2, no. 5, pp. 347–380, 2003.

[13] A. Abdollahi and M. Pedram, "Analysis and synthesis of quantum circuits by using quantum decision diagrams," in *Design, Automation and Test in Europe*, 2006, pp. 317–322.

[14] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo, "An XQDD-based verification method for quantum circuits," *IEICE Transactions*, vol. 91-A, no. 2, pp. 584–594, 2008.

[15] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, "QMDDs: Efficient quantum function representation and manipulation," *IEEE Trans. on CAD*, vol. 35, no. 1, pp. 86–99, 2016.

[16] M. Soeken, R. Wille, C. Hilken, N. Przigoda, and R. Drechsler, "Synthesis of reversible circuits with minimal lines for large functions," in *Asia and South Pacific Design Automation Conf.*, 2012, pp. 85–92.

[17] P. Niemann, R. Wille, and R. Drechsler, "Efficient synthesis of quantum circuits implementing Clifford group operations," in *Asia and South Pacific Design Automation Conf.*, 2014, pp. 483–488.

[18] A. Zulehner and R. Wille, "One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic," *IEEE Trans. on CAD*, vol. 37, no. 5, pp. 996–1008, 2018.

[19] P. Niemann, R. Wille, and R. Drechsler, "Improved synthesis of Clifford+T quantum functionality," *Design, Automation and Test in Europe*, 2018.

[20] R. Wille, D. Große, D. M. Miller, and R. Drechsler, "Equivalence checking of reversible circuits," in *Int'l Symp. on Multi-Valued Logic*, 2009, pp. 324–330.

[21] S. Yamashita and I. L. Markov, "Fast equivalence - checking for quantum circuits," *Quantum Information & Computation*, vol. 10, no. 9&10, pp. 721–734, 2010. [Online]. Available: http://www.rintonpress.com/xxqic10/qic-10-910/0721-0734.pdf

[22] P. Niemann, R. Wille, and R. Drechsler, "Equivalence checking in multi-level quantum systems," in *Conference on Reversible Computation*, 2014, pp. 201–215.

[23] L. Burgholzer and R. Wille, "Improved DD-based equivalence checking of quantum circuits," in *Asia and South Pacific Design Automation Conf.*, 2020, pp. 127–132.

[24] A. Zulehner and R. Wille, "Advanced simulation of quantum computations," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 848–859, 2019. [Online]. Available: https://doi.org/10.1109/TCAD.2018.2834427

[25] ——, "Matrix-vector vs. matrix-matrix multiplication: Potential in DD-based simulation of quantum computations," in *Design, Automation and Test in Europe*, 2019, pp. 90–95.

[26] A. Zulehner, P. Niemann, R. Drechsler, and R. Wille, "Accuracy and compactness in decision diagrams for quantum computation," in *Design, Automation and Test in Europe*, 2019.

[27] R. Drechsler and B. Becker, "Ordered Kronecker functional decision diagrams-a data structure for representation and manipulation of boolean functions," *IEEE Trans. on CAD*, vol. 17, no. 10, pp. 965–973, 1998. [Online]. Available: https://doi.org/10.1109/43.728917

[28] R. E. Bryant and Y. Chen, "Verification of arithmetic circuits with binary moment diagrams," in *Design Automation Conf.*, 1995, pp. 535–541. [Online]. Available: http://doi.acm.org/10.1145/217474.217583

[29] P. Niemann, R. Wille, and R. Drechsler, "On the "Q" in QMDDs: Efficient representation of quantum functionality in the QMDD data-structure," in *Conference on Reversible Computation*, 2013, pp. 125–140. [Online]. Available: https://doi.org/10.1007/978-3-642-38986-3_11

[30] S. Minato, "Zero-suppressed BDDs for set manipulation in combinational problems," in *Design Automation Conf.*, 1993, pp. 272–277.

[31] F. Somenzi, "CUDD: CU decision diagram package release 3.0.0," http://vlsi.colorado.edu/%7Efabio/, 2015.

[32] D. E. Knuth, "The art of computer programming: Binary decision diagrams," https://www-cs-faculty.stanford.edu/%7Eknuth/programs.html, 2011.

[33] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, "Simulation of electronic structure hamiltonians using quantum computers," *Molecular Physics*, vol. 109, no. 5, pp. 735–750, 2011.

[34] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, "A new universal and fault-tolerant quantum basis," *Information Processing Letters*, vol. 75, no. 3, pp. 101–107, 2000.

[35] S. Lang, *Cyclotomic Fields I and II*. Springer, 1990.

[36] M. Artin, *Algebra, Second Edition*. Pearson Prentice Hall, 2011.

[37] F. S. Foundation, "The gnu multiple precision arithmetic library release 6.1.2," https://gmplib.org/, 2016.

[38] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, "Exponential algorithmic speedup by a quantum walk," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, 2003, pp. 59–68. [Online]. Available: http://doi.acm.org/10.1145/780542.780552

[39] A. S. Green, P. L. Lumsdaine, N. J. Ross, P. Selinger, and B. Valiron, "Quipper: a scalable quantum programming language," in *Conference on Programming Language Design and Implementation*, 2013, pp. 333–342. [Online]. Available: http://doi.acm.org/10.1145/2462156.2462177

[40] "QMDD package," http://www.informatik.uni-bremen.de/agra/eng/qmdd.php.

Philipp Niemann (M'15) received a Diploma degree in mathematics and a PhD degree in computer science from the University of Bremen, Germany, in 2012 and 2016, respectively. There, he worked in the Group for Computer Architecture under the supervision of Prof. Dr. Rolf Drechsler and Prof. Dr. Robert Wille. In 2017, he joined the Cyber-Physical Systems (CPS) department at the German Research Center for Artificial Intelligence (DFKI) in Bremen. His research interests are in the design of reversible and quantum circuits with a focus on decision diagrams as well as in the verification of formal models. Philipp Niemann published several papers on international conferences such as ASP-DAC, DATE, RC, MEMOCODE, and MoDELS.

Alwin Zulehner (S'17) received his MSc degree as well as his PhD in computer science from the Johannes Kepler University Linz, Austria in 2015 and 2019, respectively. His research interests include design automation for quantum computing. In this area, he has published several papers at international conferences and journals such as the IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems (TCAD), the Asia and South Pacific Design Automation Conference (ASP-DAC), the Design, Automation and Test in Europe (DATE) conference, the Design Automation Conference (DAC), and the International Conference on Computer-Aided Design (ICCAD). For his work, he got awarded with Best Student Awards as well as the EDAA Outstanding Dissertation Award and won the IBM Qiskit Challenge for his development of a quantum compiler approach.

Rolf Drechsler (M'94-SM'03-F'15) received the Diploma and Dr. phil. nat. degrees in computer science from the Johann Wolfgang Goethe University in Frankfurt am Main, Germany, in 1992 and 1995, respectively. He worked at the Institute of Computer Science, Albert-Ludwigs University, Freiburg im Breisgau, Germany, from 1995 to 2000, and at the Corporate Technology Department, Siemens AG, Munich, Germany, from 2000 to 2001.

Since October 2001, Rolf Drechsler is Full Professor and Head of the Group of Computer Architecture, Institute of Computer Science, at the University of Bremen, Germany. In 2011, he additionally became the Director of the Cyber-Physical Systems Group at the German Research Center for Artificial Intelligence (DFKI) in Bremen. His current research interests include the development and design of data structures and algorithms with a focus on circuit and system design. He is an IEEE Fellow. From 2008 to 2013 he was the Vice Rector for Research and Young Academics at the University of Bremen. Since 2018 he is the Dean of the Faculty of Mathematics and Computer Science.

Rolf Drechsler was a member of Program Committees of numerous conferences including e.g., DAC, ICCAD, DATE, ASP-DAC, FDL, MEMOCODE, and FMCAD. He was Symposium Chair at ISMVL 1999 and 2014, and the Topic Chair for "Formal Verification" at DATE 2004, DATE 2005, DAC 2010, and DAC 2011 and 2018.

He received best paper awards at the Haifa Verification Conference (HVC) in 2006, the Forum on specification & Design Languages (FDL) in 2007 and 2010, the IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS) in 2010 and the IEEE/ACM International Conference on Computer-Aided Design (ICCAD) in 2013 and 2018. Rolf Drechsler is an Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Very Large Scale Integration Systems, ACM Journal on Emerging Technologies in Computing Systems, and further journals.

Robert Wille (M'06–SM'15) is Full Professor at the Johannes Kepler University Linz, Austria. He received the Diploma and Dr.-Ing. degrees in Computer Science from the University of Bremen, Germany, in 2006 and 2009, respectively. Since then, he worked at the University of Bremen, the German Research Center for Artificial Intelligence (DFKI), the University of Applied Science of Bremen, the University of Potsdam, and the Technical University Dresden. Since 2015, he is working in Linz. His research interests are in the design of circuits and systems for both conventional and emerging technologies. In these areas, he published more than 300 papers in journals and conferences and served in editorial boards and program committees of numerous journals/conferences such as TCAD, ASP-DAC, DAC, DATE, and ICCAD. For his research, he was awarded, e.g., with Best Paper Awards (e.g., at ICCAD), a DAC Under-40 Innovator Award, a Google Research Award, and more.