Automated Reasoning in Quantum Circuit Compilation

Dimitrios Thanos, Alejandro Villoria, Sebastiaan Brand, Arend-Jan Quist
Jingyi Mei, Tim Coopmans and Alfons Laarman

Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden 2333 CA, The Netherlands

Abstract. Automated reasoning techniques have been proven of immense importance in classical applications like formal verification, circuit design and probabilistic inference. The domain of quantum computing poses new challenges of a different nature, such as the compilation of quantum circuits, which involves "quantum-hard" tasks such as the simulation, optimization, synthesis, and equivalence checking of quantum circuits. We ask the question of how effective the methods motivated by classical automated reasoning can be for quantum compilation. We assess their current applicability to this new domain by discussing the recent advances. In particular, we focus on three core automated reasoning approaches: decision diagrams, satisfiability and graphical calculus-based methods. In this survey, we explain in a manner accessible to those unfamiliar with quantum computing concepts how these prominent automated reasoning methods have found numerous applications in quantum circuit compilation. We find that surprisingly all considered reasoning methods, while originally developed for classical purposes, can excel at various compilation tasks for even universal quantum circuits.

1 Towards Quantum Supremacy

Quantum supremacy [137] is the question of obtaining the first example of a problem for which quantum computers provably surpass classical computers in theory and/or practice. There are various reasons to pursue this challenge [193].

First, technology advance has led to the miniaturization of classical computers, rendering them powerful and cost-effective. However, this trend has now extended into the micro-level where quantum phenomena come into play, causing insurmountable hurdles for further miniaturization. Alternatively, the embrace of quantum effects could lead to further miniaturization and innovation.

Second, quantum computing (QC) holds the promise of revolutionizing the field of computation by surpassing classical computers in terms of efficiency, particularly in tackling tasks that are deemed classically intractable [121, 113]. Quantum algorithms can leverage quantum phenomena like entanglement and constructive interference to tackle problems beyond the reach of classical computers. For example, for period finding, Shor's algorithm features a performance exponentially faster than their best-known classical versions [121].

Third, theoretical computer science aims to understand the strengths and limitations of the most powerful computers that nature allows. So it makes sense to focus on studying the potential of quantum computers which are closer to the limit of our current understanding of nature. Physicists, chemists, and other scientists are constantly dealing with quantum-hard problems like Hamiltonian simulation, computing ground state energies, etc. This is where Feynman's conception of the quantum computer originated [104].

However, despite progress [6, 89], quantum supremacy for useful problems remains elusive. A large-scale quantum computer with sufficient error-correction is yet to be built. On the theory side, one cannot rule out the possibility that the runtime of certain quantum algorithms gets matched by new classical algorithms. This has happened before. The quantum recommender algorithms were thought to be exponentially faster than their classical counterparts, however, Tang [163] showed (at the age of 18) that there exist classical algorithms with similar asymptotic performance.

Nonetheless, a good reason to remain optimistic about this kind of research is that challenges in quantum circuit compilation are of exactly the same nature as challenges with which physicists and quantum chemists struggle, as expressed in the third point above. (Not to mention that Tang's proof has led to a class of improved, quantum-inspired, classical algorithms.) Therefore, progress in QC is progress in fundamental research that has the potential to advance our understanding of the many quantum-hard problems that nature confronts us with.

Gate-based quantum computing, one of the most prevalent models of quantum computing, involves the utilization of a limited set of quantum gates, specifically reversible operators designed to manipulate qubits. These operators form quantum circuits, which are not necessarily unique, meaning that different circuits that implement the same computation can exist. In the current era of noisy intermediate-scale quantum computing (NISQ) [138], there are many challenges that we need to overcome when compiling quantum circuits into real-world devices. Such challenges are the high noise levels, the shallow depth of the circuits that can be practically implemented, and the various constraints (connectivity, topology, native gate sets, etc.) [65, 46]. It is evident that circuit compilation problems form an important hurdle on the road to achieving quantum supremacy.

A promising range of techniques for addressing these questions exists within the field of automated reasoning. In the analysis of (classical) system behavior, computer scientists are often dealing with a combinatorial explosion. As a consequence, many powerful formalisms and approaches were developed to reason about such systems. For instance, decision diagrams [29, 3], satisfiability [21] and theorem provers [24, 19, 130, 99, 115], offer rigorous techniques to verify the behavior of classical systems and ensure their accuracy and dependability.

The state of n quantum bits is generally represented as 2^n complex values [121]. Consequently, already the simulation of quantum circuits, a core task in circuit compilation, as we will see, must tackle a combinatorial explosion similar to that encountered when analyzing the behavior of classical systems. Hence, many of the techniques traditionally used for the analysis of classical sys-

tems in the field of automated reasoning and formal methods have been proven useful for quantum circuit compilation. For instance, decision diagrams have been established as efficient analysis and optimization tools for quantum circuits [199, 187], model counting shows promise for simulation [107] and equivalence checking [108], satisfiability for analysis of non-universal quantum circuits [17, 189, 195], satisfiability modulo theories for circuit verification [12, 39] and equivalence checking [4, 5], and theorem-prover-style deductive approaches for simulation and equivalence checks [131, 59, 185]. In the context of quantum computing, the most prominent type of theorem provers is graphical calculi, so we will narrow the discussion about theorem provers to graphical calculi.

This survey aims to comprehensively document the progress of the transfer of 'classical' automated reasoning tools to the relatively new field of quantum computing. The following methodology was utilized to define its scope. We concentrate on the following three core automated reasoning approaches, focusing on some of the most prominent ones employed for quantum circuit compilation.

- Decision diagrams;
- Satisfiability (SAT / SMT / #SAT);
- Graphical calculus-based methods

For each, we used Google Scholar to identify the ten most relevant publications applying the particular method to a task in quantum compilation, as defined in Section 3 to include the simulation, synthesis, optimization, and equivalence checking of quantum circuits. We use the search terms "decision diagrams" / "satisfiability / "SAT" / "SMT" / "Graphical calculi" and "quantum circuit". The publications outside the scope of our paper are discarded. Starting from these works, we did a literature review following the trace of citations. We explicitly exclude formal verification, such as model checking and Hoare logic-based deduction, from our queries, as we are mainly interested in how automated reasoning methods transfer onto progressing the quantum supremacy challenge and not how formal verification can be lifted to the quantum domain. In this, our survey differs substantially from earlier surveys [38, 100], which focus on correctness verification of quantum circuits and algorithms.

The text is structured as follows. We first introduce quantum computing in more detail in Section 2. We then define what we mean by quantum circuit compilation by discussing the challenging tasks it encompasses in Section 3. In Section 4-6, we discuss in detail how the three automated reasoning approaches found applications in quantum circuit compilation. We also report on automated reasoning that superseded the particular version tailored to quantum computing, finding for instance an interesting parallel in the early invention of decision diagrams representing pseudo-Boolean functions (i.e., quantum states) and their later development in quantum computing.

In Section 7, we also discuss other methods that we could identify as used for quantum circuit compilation (some drawn from physics, such as tensor networks or path integral-based methods). We conclude in Section 8 that automated reasoning methods originally developed for classical problems also excel in various compilation tasks and will likely find more applications in quantum computing.

2 Fundamental Concepts of Quantum Computing

In this section we briefly explain the core concepts of quantum computing. The goal is not to give a complete overview of all of the underlying mathematics, but rather to familiarize the reader with some of the concepts and terms used. For a comprehensive overview of quantum computing, we refer the reader to [121].

Quantum states. Where classical states are described by bits, quantum states are described by quantum bits (qubits). A qubit, just like a classical bit, can be in a state 0 or 1, described by vectors $|0\rangle = \begin{bmatrix}1 & 0\end{bmatrix}^\intercal$ and $|1\rangle = \begin{bmatrix}0 & 1\end{bmatrix}^\intercal$. Unlike classical bits, qubits can be in *superposition*, described by a linear combination of the $|0\rangle$ and $|1\rangle$ states. Concretely, the state of a qubit is given by $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{bmatrix}\alpha_0 & \alpha_1\end{bmatrix}^\intercal$, with $\alpha_0, \alpha_1 \in \mathbb{C}$ and $|\psi\rangle$ a unit vector (i.e. $|\alpha_0|^2 + |\alpha_1|^2 = 1$).

Another key difference between classical bits and qubits is how multi-(qu)bit systems are composed. Whereas as n-bit state can simply be described by a bit string $b \in \{0,1\}^n$, multi-qubit states are composed from single-qubit states by means of the tensor product. Intuitively, this tensor product is similar to the Cartesian product of sets. Formally, the tensor product of an $r_a \times c_a$ matrix A and an $r_b \times c_b$ matrix B yields a matrix $A \otimes B$ of dimension $r_a r_b \times c_a c_b$ equal to

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1c_a}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{r_a 1}B & A_{r_a 2}B & \cdots & A_{r_a c_a}B \end{bmatrix}.$$

For example, for two single-qubit states $|\psi_A\rangle = \begin{bmatrix}\alpha_0 & \alpha_1\end{bmatrix}^\intercal$ and $|\psi_B\rangle = \begin{bmatrix}\beta_0 & \beta_1\end{bmatrix}^\intercal$, we get that $|\psi_A\rangle \otimes |\psi_B\rangle = \begin{bmatrix}\alpha_0\beta_0 & \alpha_0\beta_1 & \alpha_1\beta_0 & \alpha_1\beta_1\end{bmatrix}^\intercal$. In general, the state of n qubits is given by a 2^n-dimensional vector. For convenience, we define $|b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes ... \otimes |b_n\rangle$ for $|b\rangle \in \{0,1\}^n$. Observe that $|b\rangle = e_b$, i.e., a 2^n-length vector with only index $b \in \{0,1\}^n$ set to 1 and the other entries to 0.

Similar to the conjugate of a complex number, a quantum state $|\varphi\rangle = \begin{bmatrix}\alpha_0 & \alpha_1 & ... & \alpha_{2^n}\end{bmatrix}^\intercal$ has an 'adjoint' $\langle\varphi| = (|\varphi\rangle^*)^\intercal = \begin{bmatrix}\alpha_0^* & \alpha_1^* & ... & \alpha_{2^n}^*\end{bmatrix}$, i.e., its transpose, a row vector, with all complex entries conjugated. Observe that $\langle b|\varphi\rangle = \alpha_b$. All states must have unit length, which we can now formulate as $\langle\varphi|\varphi\rangle = 1$.

It is important to note that some joint states cannot be decomposed as a tensor product of smaller states. Such states are *entangled*. For example, states $|\psi_3\rangle = 1/\sqrt{2}(|00\rangle + |11\rangle) = \begin{bmatrix}1/\sqrt{2} & 0 & 0 & 1/\sqrt{2}\end{bmatrix}^\intercal$ and $|\psi_4\rangle$ from Figure 1 are entangled.

Quantum operations. There are two types of operations on quantum states: gates, and measurements. Quantum gates are linear maps that are information-preserving (i.e., reversible) and norm-preserving. This makes it so that an n-qubit quantum gate U is given by an $2^n \times 2^n$ unitary matrix, where unitarity is defined as $UU^\dagger = U^\dagger U = I$ with $U^\dagger = (U^*)^\intercal$ denoting the conjugate transpose of U.

The effect of a gate (matrix) on a qubit state (vector) is computed through matrix-vector multiplication. A sequence of quantum gates acting on a state is typically visualized in a quantum circuit (Figure 1). Some examples of quantum gates are the Pauli gates (X, Y, Z) and Clifford gates (H, S, CNOT).

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

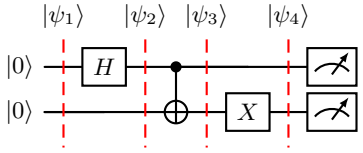
To give an intuition, the X gate acts as a classical bit-flip and the Z and S gates as a 'phase-flip' on $|1\rangle$, yielding $-|1\rangle$ and $i|1\rangle$ respectively, the Hadamard (H) gate brings a qubit from the $|0\rangle$ state into a *uniform superposition* of $|0\rangle$ and $|1\rangle$, and the two-qubit controlled-not gate (CNOT) acts as a bit flip on a 'target' only when the control qubit is one. Combined with the Hadamard, the CNOT gate can produce entangled states (as in Figure 1).

We say that two unitaries U and V are equivalent up to 'global phase' if $U = cV$, where $c \in \mathbb{C}$. The term 'global phase' refers to the complex factor c, which does not affect any observable properties of unitaries [121].

A (standard) measurement of a qubit in the state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ collapses the qubit onto the $|0\rangle$ ($|1\rangle$) state, with probability equal to $|\alpha_0|^2$ ($|\alpha_1|^2$). For measuring a single qubit within a multi-qubit state $|\varphi\rangle$, we note that it can always be written as $|\varphi\rangle = \alpha_0|0\rangle \otimes |\varphi_0\rangle + \alpha_1|1\rangle \otimes |\varphi_1\rangle$, where $|\alpha_0|^2$ and $|\alpha_1|^2$ again correspond to the probabilities of collapsing to the $|0\rangle$ and $|1\rangle$ states.

The stabilizer formalism. The final concept we introduce is the stabilizer formalism [69], which describes a class of quantum circuits and corresponding quantum states that are known to be classically simulatable in polynomial time.

The circuits in question are those composed of Clifford gates, which are all quantum gates that can be generated (under multiplication and the tensor product) from H, S, and CNOT. The stabilizer states are all states that can be produced by a Clifford circuit initialized to the all-zero state $|0\rangle^{\otimes n}$. Any stabilizer state can be represented by a set of n "stabilizers". In this context, a matrix $P \in \{\pm P_1 \otimes ... \otimes P_n \mid P_i \in \{X, Y, Z, I\}\}$ is a stabilizer of an n-qubit state $|\sigma\rangle$ if $|\sigma\rangle$ is an eigenvector of P, i.e. $P|\sigma\rangle = \pm|\sigma\rangle$. Tracking these n stabilizers (including the $\pm$ sign) can be done with $2n^2 + n$ Boolean values, which are typically encoded in a so-called stabilizer tableau.



$$|\psi_1\rangle = |0\rangle \otimes |0\rangle = |00\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\intercal$$
$$|\psi_2\rangle = (H \otimes I)|\psi_1\rangle = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \end{bmatrix}^\intercal$$
$$|\psi_3\rangle = \text{CNOT}|\psi_2\rangle = \begin{bmatrix} 1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} \end{bmatrix}^\intercal$$
$$|\psi_4\rangle = (I \otimes X)|\psi_3\rangle = \begin{bmatrix} 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \end{bmatrix}^\intercal$$

Fig. 1. An example 2-qubit quantum circuit. Each qubit is represented by a horizontal wire, and operations are applied from left to right. As is common, we write $|xy\rangle$ as shorthand for $|x\rangle \otimes |y\rangle$. Measuring both qubits after obtaining $|\psi_4\rangle = 1/\sqrt{2}(|01\rangle + |10\rangle)$ gives $|01\rangle$ or $|10\rangle$ each with probability $|1/\sqrt{2}|^2 = 1/2$.

Moreover, this tableau can be efficiently updated to implement the semantics of any Clifford gate. The fact that stabilizer states are representable by a succinct and easy-to-update representation makes Clifford circuits classically simulatable.

It is important to note that while stabilizer states play an important role in quantum computing, such as in quantum error correction [78, 164, 16, 98] and measurement-based quantum computing [141], they are by themselves not sufficient for universal quantum computation. Generalizations of the stabilizer formalism further allow a complete discretization of the (universal) quantum state space as we shall see in subsequent sections.

By adding a single non-Clifford gate, like a T gate or any arbitrary rotation gate R_X, R_Y, R_Z, to the Clifford gate set, we obtain universal quantum computing [120]. Stabilizer-rank-based methods [28] allow (classical) fixed-parameter tractable simulation in the number of T gates in the circuit.

3 Challenges of Quantum Circuit Compilation

Quantum circuit compilation is the process of mapping an instance of a quantum algorithm to a specific quantum hardware, i.e., a specific quantum processing unit (QPU). A more efficient mapping can bring us closer to quantum supremacy, especially when we aim to identify useful problems at which quantum computers excel, since they are likely to require structured circuits instead of randomized ones [6]. As instances, we will consider circuits, and not programming languages (which can be translated to circuits), since we are interested in suing automated reasoning for hard problems. There are multiple problems crucial to quantum circuit compilation. For this text, we focus on the following tasks.

1. **Circuit Simulation**: Classical simulation of quantum circuits enables basic analyses, but, as we shall see, is also often a sub-task in other compilation tasks. Better simulation methods arguably translate into improvement in other compilation tasks. We can distinguish two types of simulation.
   (a) **Strong Circuit Simulation**: Given a quantum circuit, and a computational basis state, compute the probability of measuring that state.
   (b) **Weak Circuit Simulation**: Given a quantum circuit, sample from the probability distribution of its measurement outcomes.[1]
2. **Circuit Optimization**: Given a quantum circuit and a set of hardware constraints, produce another circuit that represents an equivalent computation satisfying the constraints. The new circuit should match the hard constraints posed by the QPU, such as topology (i.e., connectivity between qubits), as well as soft constraints that allow the reduction of execution time and noise exposure introduced by imperfections in the QPU. For example, certain gates might be more expensive in terms of noise or entangling certain pairs of qubits could be more error-prone [10]. Tasks such as explicitly reducing the number of gates in a circuit fall in this category [158], as well as other tasks such as layout synthesis and qubit mapping and routing [112],

[1] Measurements can always be deferred until the end of the circuit [121, §4.4].

which consist of mapping a logical circuit into a hardware-aware one that satisfies the physical qubit-connectivity constraints.

3. **Circuit Synthesis**: Given a specification, produce a quantum circuit that adheres to it [58]. This specification can be relational, e.g., another circuit or a Hoare logic expression [196], or simply an input and output state, in which case, we speak more specifically of **quantum state preparation**. The specification can also be a unitary, in which case we speak of 'decomposition'. Synthesis is often combined with optimization. In that case, the specification is extended with the soft and hard constraints discussed above.
4. **Circuit Equivalence Checking**: Given two circuits, decide whether they represent equivalent unitaries. Since circuit compilation tasks modify circuits, having a method for checking the correctness of the result is essential.

Although we have categorized the tasks into four main categories, it is important to note that they come in a whole spectrum. For example, there are many hardware constraints under which one can optimize and trade-offs come into play. Moreover, a heuristic solution is sometimes good enough for soft constraints. While we are mainly interested in sound and complete methods, since they match the capabilities of the considered automated reasoning techniques, we also include heuristic solutions built on automated reasoning.

Each compilation task can be defined as an exact or approximate problem (i.e., with bounded error like in the classical class BPP). For instance, exact equivalence checking requires that the unitaries are equal up to global phase (see Section 2), while approximate checking merely requires that the two unitaries always produce the same states up to a certain *fidelity* [103, 82] (a measure for 'closeness' of two quantum states [121]). The complexity classes for exact problems depend strongly on the gate set, because the gate set determines the reachable states [67]. On the other hand, the bounded-error complexity classes, where BQP is the quantum analog of P and QMA the quantum analog of NP [94], are invariant under the gate set (a motivation behind their definition). The exact version is harder, e.g., exact strong simulation is already #P-complete [120, 88]. Nonetheless, exact reasoning can be appealing because it allows discretization.[2] Surprisingly, exact reasoning methods are even used to compute the approximate versions of these compilation tasks. For example, [184, 82] even solve the "quantum NP"-hard circuit equivalence, which is QMA-hard [84] to approximate and NQP-hard [162] to compute exactly. Exact reasoning also allows linear #SAT encodings of simulation and equivalence checking [107, 108] (see Section 5) and is used extensively in ZX-calculus (see Section 6).

In Section 4–6, we detail the applications of three main automated reasoning approaches in quantum circuit compilation. We focus on the historical context and the technical aspects that have influenced the adaptation and evolution of these automated reasoning methods in this new application area. This illustrates how quantum circuit compilation benefited from automated reasoning techniques originally developed for reasoning about classical systems.

[2] With a ring discretizing all complex numbers calculable in a given gate set [92, 67].

4 Decision Diagrams

To point out parallels in the early development of decision diagrams and their later use in quantum computing, we order this section mostly chronologically.

Historical Background. Akers [3] defined the binary decision diagram (BDD) for compactly representing a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. Bryant [29] established the importance of BDDs by giving efficient manipulation algorithms that can compute $F \circ G$ where F, G are any functions represented as BDDs and $\circ$ is any Boolean operator. Using Bryant's work as a basis, operations have been invented for quantification [106], matrix multiplication [66, 56], and even reachability [25]. With the multi-terminal [40] or algebraic decision diagram (ADD) [9], BDDs were generalized to support the representation of pseudo-Boolean function $f\colon \{0,1\}^n \to D$ for any (ring-structured) domain $D \in \{\mathbb{R}, \mathbb{C}, ...\}$.

The main idea that led to the construction of decision diagrams can be traced back to Boole and his identity formula $f = (x \wedge f_x) \vee (\overline{x} \wedge f_{\overline{x}})$ [22], where $f_x, f_{\overline{x}}$ are the sub-functions with variable x restricted to $1, 0$ respectively. This is the expansion identity (also known as decomposition identity or Shannon decomposition) and is often attributed to Shannon since its appearance in his paper on

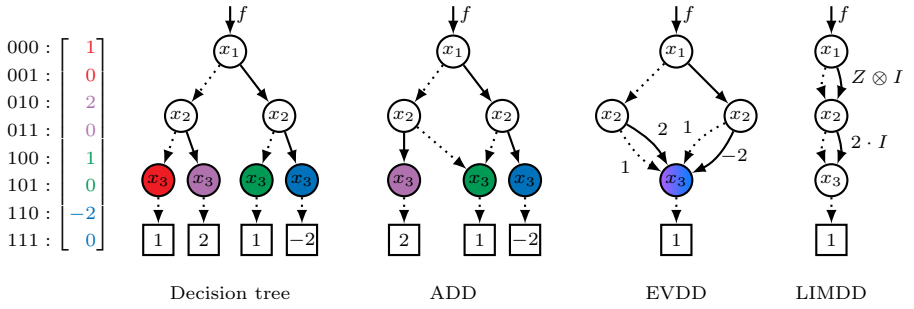


Fig. 2. A pseudo-Boolean function $f(x_1, x_2, x_3) = \overline{x_3} \cdot (2 \cdot x_2 \cdot (\overline{x_1} - x_1) + \overline{x_2})$ represented as (non-normalized quantum state) vector and as an (exponentially-sized) decision tree. A node labeled with variable x represents a Shannon decomposition on x. An outgoing dashed edge represents the sub-functions where the variable x is restricted to false (0) and a solid edge where the variable is restricted to true (1). The boxes represent leaves. We omit zero leaves.
From a decision tree to algebraic decision diagram (ADD): The red node is merged into the green as both represent the same sub-function $f_{00}(x_3) = f_{10}(x_3) = 1 - x_3$.
From algebraic decision diagram to edge-valued decision diagram (EVDD): The purple, green, and blue nodes represent sub-functions that are equivalent up to a constant factor of 2, 1, -2 respectively. By putting the factors on the edges, these equivalent nodes can be merged. Unlabeled edges have implicit factor 1.
From edge-valued decision diagram (EVDD) to LIMDD: The Pauli LIM $Z \otimes I$ maps f_0 to f_1 (as vectors, i.e. $(Z \otimes I) \cdot [\, f_{100}\ f_{101}\ f_{110}\ f_{111}\,]^T = [\, f_{100}\ f_{101}\ -f_{110}\ -f_{111}\,]^T = f_0$). Both x_2 nodes can thus be merged by putting this map on the (high) edge. Unlabeled edges to nodes x_i have implicit map $1 \cdot I^{\otimes i}$.

classical circuits [152]. This decomposition can be viewed as "peeling off" a single variable x. By iterating this process on sub-functions, we can build the typical exponential binary tree ending with $0, 1$ leaves as illustrated in Figure 2 for a pseudo-Boolean function. Note that this results in a total variable order along a path in the decision diagram: All decision diagrams discussed here fix a single variable order along all paths as this crucially enables the efficient manipulation operations given by Bryant [29]. When decision diagram nodes in this binary tree represent the same sub-function, e.g., two leaf nodes with same value, they are merged. For many practical functions, such as those arising in verification problems, this node merging can result in significant size reductions.

Lai, Pedram and Vrudhula [97] first showed that the ADD structure can be made more succinct by merging nodes that represent equivalent sub-functions up to a constant additive factor, which is then placed on the edges. Tatershofer and Pedram [159, 160, 179] improved on this by also allowing multiplicative constants p apart from the additive constants a, i.e., affine transformations $p \cdot f + a$. By requiring the domain D to be a semi-ring, Wilson [191] defined the semi-ring labeled decision diagram (SLDD) that factors out only multiplicative constants. In line with the early inventors, we call all these structure "edge-valued decision diagrams" (EVDDs). Figure 2 illustrates how an ADD is compacted as an EVDD, with the factored out multiplicative constants on the edges. Finally, Sanner and McAllester [147] developed affine ADDs (AADDs), and Fargier, Marquis and Schmidt [62] generalized and related various of the above decisions diagrams.

As in all data structure design, there is a trade-off between succinctness and efficiency of operations. Darwiche [52] first mapped these trade-offs for different data structures representing Boolean functions. His 'knowledge compilation map' shows, inter alia, that, for BDD, basic Boolean operations are efficient, but more complicated operations like unbounded quantification are not, which was already known from the fact that poly-time matrix multiplication with BDDs would imply $\mathsf{P} = \mathsf{NP}$ [106] and that reachability using BDDs is PSPACE-hard [63].

Later, Fargier et al. [62] showed that more or less similar results hold for manipulation operations on ADDs, replacing Boolean operations for point-wise addition, multiplication, and min/max computations in the pseudo-Boolean domain. However, they also show that basic operations, such as point-wise addition, become intractable for EVDD and AADD. This affects the implementation of the Hadamard gate, as we discuss in the next section. On the other hand, this worst-case analysis obfuscates the fact that there are no functions for which AADD or EVDD is slower on point-wise addition than ADD [147]. This is because the intractability for AADD / EVDD addition only occurs when the corresponding ADD would already be large for representing the input functions.

The structural differences between these different decision diagrams are summarized in Table 1, which also summarizes the chronology, includes references and the quantum-inspired decision-diagram versions discussed next.

Table 1. Various decision diagrams (DDs) used in the literature (extended from [177]). The column "node merge" lists the conditions under which two decision diagram nodes, representing functions f and g, are merged. Here, $p, a \in \mathbb{C}$ are complex constants, $P = P_1 \otimes ... \otimes P_n$ a sequence of single-qubit Pauli gates P_i, and $f + a$ means the function $f(\vec{x}) + a$ for all $\vec{x}$. All DDs, except QDD, have complex scalars as terminals and their internal nodes v, w thus represent functions $f, g\colon \{0,1\}^n \to \mathbb{C}$. In QDD, the terminal node represents the $|0\rangle$ vector, and thus QDD can be seen as functions $f, g\colon \{0,1\}^{n-1} \to \mathbb{C}^2$. Here R_X is a single-qubit rotation operator. All these decision diagrams use a fixed total variable order. CFLOBDD is not in the table as it processes variables via recursive bisection.

(Quantum) decision diagrams (and variants)	**Node merge**
Decision Tree	(no merging)
MTBDD (1993) [40], ADD (1997) [9], QuiDD (2003) [169]	$f = g$
$\text{SLDD}_{\times}$ (2004) [191], QMDD (2006) [111], XQDD (2008) [183], TDD (2021) [83]	$f = p \cdot g$
EVBDD (1994) [97], SLDD_{+} (2005) [62]	$f = g + a$
FEVBDD (1994) [159, 160, 179], $\text{SLDD}_{+,\times}$ (2005) [62], AADD (2005) [147]	$f = p \cdot g + a$
QDD (2006) [2]	$f = R_X \cdot g$
LIMDD (2023) [176, 178, 177]	$f = p \cdot P \cdot g$

Decision Diagrams in Quantum Circuit Compilation. Decision diagrams have been pioneered for the efficient representation and manipulation of quantum states by Viamontes in the form of QuiDDs [169], which are essentially ADDs with a complex domain $D = \mathbb{C}$. The basic insight is that a quantum state $|\varphi\rangle$ can be viewed as a pseudo-Boolean function $f_\varphi(\vec{x}) = \langle\vec{x}|\varphi\rangle$, from computational basis states $\vec{x}$ to complex amplitudes $\langle\vec{x}|\varphi\rangle$. QuiDDs have been used for simulation of quantum computing [170, 169, 171].

QuiDDs were succeeded by QMDDs [111], which can be viewed as edge-valued decision diagrams on the domain of complex numbers, transferring the compactness of EVDD to the application of quantum circuit compilation. Here it should be noted that the (pointwise) addition operation is required for the implementation of the Hadamard gate, which can result in a possible blowup of the diagram, which nonetheless does not exceed the size of the corresponding QuiDDs, as discussed above for ADD and EVDD. The tractability of gate implementations for QMDD is presented in [176, 177], showing that all other primitive gate operations are tractable for QMDD, except for the swap gate.

QMDD has been applied to many quantum circuit compilation tasks: quantum circuit simulation [199], equivalence checks [122, 33, 31, 32], including approximate equivalence checks [184], and synthesis [198, 123]. QMDDs have also been used to simulate Hamiltonians [144, 80] and circuits with noise [73, 74, 72].

QDD [2] offers an interesting variation on QMDD, as Table 1 illustrates. Further, while QMDD represents unitaries with quaternary DD nodes, TDD [83] clones each qubit variable as was traditionally done [66, 106, 159].

Interestingly, certain families of stabilizer states yield an exponentially-sized representation in QMDD as shown in [176]. This is surprising given the importance of stabilizer states and that simulation (and representation) of stabilizer states is tractable as discussed in Section 2. LIMDD [176, 178] generalizes QMDD

to solve this, reducing the required space in worst case from $\Omega(2^{\sqrt{n}})$ to $\mathcal{O}(n^2)$ for all stabilizer states. LIMDD achieves this feat by merging nodes that are equivalent up to tensor products of single-qubit Pauli operators (as well as constant complex factors, such as in QMDDs). Consequently, where a QMDD node has decomposition $|0\rangle\otimes\alpha_0\,|\varphi_0\rangle+|1\rangle\otimes\alpha_1\,|\varphi_1\rangle$ pointing to a node φ_0 (φ_1) with the low (high) edge which is labelled with complex number α_0 (α_1), a LIMDD replaces the complex numbers α_0, α_1 by tensor products of Pauli matrices $P = P_1\otimes...\otimes P_n$ times a complex number. An example is given in Figure 2, showing that some nodes which could not be merged in QMDD are equivalent in LIMDD.

The price paid for its succinctness is however that LIMDD must compute the stabilizer group for each node to achieve canonicity, causing cubic factor overhead in practice [176]. Asymptotic analysis [177] also shows that computing fidelity is also intractable for LIMDD (under common-place complexity-theoretic assumptions), while tractable in QMDD. However, Vinkhuijzen et al. [177] produced a "knowledge compilation map" for quantum information (see Figure 3) showing that only LIMDD can be more succinct than MPS and RBM (see Section 7).

Another type of decision diagram that is used for simulation is CFLOBDD. This decision diagram fundamentally differs from the decision diagrams described above, as it generalizes the linear variable order into a recursive bisection of the variables (which can be viewed as a Shannon decomposition over multiple variables at once): Every level recursively 'peels off' half of the variables instead of one variable. This is why CFLOBDD does not fit in Table 1. Moreover, a CFLOBDD has variable sharing, such that a node representing a function $f(x_1,...,x_k)$ can also be used for different variables like e.g. $f(x_{k+1},...,x_{2k})$. This

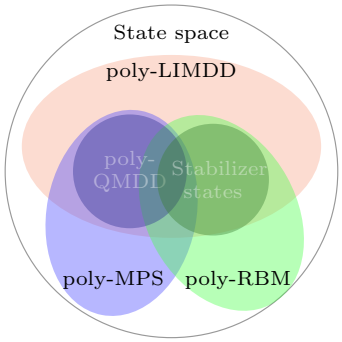


Fig. 3. The comparative succinctness of LIMDD, QMDD, MPS, and RBM (see Section 7) following from the asymptotic analysis in [177]. All these data structures, except the stabilizer formalism, can represent any quantum state (they are universal). The Venn diagram however shows families of states that can be represented in polynomial size with the different data structures. Consequently, MPS, RBM, and LIMDD are incomparable in terms of representation power: they each have their strengths and weaknesses. LIMDD is the only structure that can represent all stabilizer states and all (families of) states supported by polynomial QMDD. (Poly-)ADD and QuiDD are not shown but are strictly contained in (Poly-)QMDD.

recycling, combined with handling half of the variables in every level, results in the best case in an exponential compression with respect to BDD. For instance, the multi-qubit Hadamard matrix $H^{\otimes m}$ which can be represented by CFLOBDD in only logarithmic space. With these reductions, highly-structured quantum circuits with very many qubits can be simulated efficiently [157, 156]. The treatment of multiple variables at a time aligns with SDD [51] and 'hierarchical set decision diagram' [166], while the idea of variable shifting also was used for SDD [118]. To our knowledge, there does not yet exist a "knowledge compilation map" comparing CFLOBDD to other similar data structures.

In terms of performance, QMDD is usually among the fastest method for compilation tasks, outperforming QuiDD [199], array-based methods [199, 82], tensor networks [83, 157] (see Section 7) and ZX calculus (see Section 6) in some cases [132]. Variants like QDD and TDD have related performance [2, 83], while CFLOBDD was shown to be competitive to QMDD, SDD and tensor network [157, 155]. LIMDD, finally, proved somewhat impractical in empirical evaluation [178]. In practice, we therefore see QMDD used more, e.g. [34, 187].

There are multiple tools for the different types of decision diagrams for quantum, e.g. [186, 188, 187, 155, 178, 197, 30, 82]. It should however be pointed out that all of these use floating points by default to represent real weights on the edges. In practice, this can cause rounding errors to rapidly propagate in the discrete data structure, resulting in numerical instability [197, 124].

5 SAT/SMT-based Methods

While decision diagrams have been an indispensable data structure in many automated reasoning applications, both within as well as outside of quantum computing, they require (at worst) an exponential amount of space because they represent all satisfying assignments. While many relevant problems are known to be computationally hard in terms of time complexity, almost all of them can be solved in polynomial space. Boolean satisfiability (SAT) is the problem of deciding whether there exists a satisfying assignment to a given Boolean formula, i.e., an assignment for which the formula evaluates to true, i.e., is "satisfiable". The SAT problem is the first problem that was shown to be NP-complete [45]. Satisfiability modulo theories (SMT) generalizes SAT to other domains than Boolean, such as expressions containing bit vectors, integers or real numbers. SAT and SMT solvers are tools that tackle computationally hard problems with polynomial-space algorithms [21]. It has become common practice to solve other NP-complete problems by reducing them to SAT and then using a highly optimized solver to solve the SAT instance, the solution of which can then be translated back to the domain of the original problem. SAT-solving algorithms such as DPLL [53] and CDCL [154] traverse the exponentially large search space step-wise (thus taking only polynomial space omitting caches e.g. learned clauses), and use clever pruning heuristics to obtain good performance on many practical instances. Finally, model counting is the problem of counting

the number of satisfying assignments of a Boolean formula and model counters are tools for solving it [128, 145, 37].

SAT-based approaches have been used to tackle different types of quantum circuit synthesis problems. While SMT allows for the encoding of problems with a continuous search space [39, 12], in practice there tends to be a bias towards discretizing the problem as explained in Section 3. In quantum circuit synthesis and circuit optimization, we can identify two types of discrete problems to which SAT-based methods have been applied: layout synthesis (i.e. taking an existing circuit and remapping the qubits according to constraints) and the synthesis of discrete circuits (such as Clifford circuits and transformations of so-called "graph states"). We start with the use of SAT for simulation and equivalence checking.

Initial attempts have been made to harness the strengths of satisfiability solvers for the simulation of quantum circuits. For instance, [17] implements a simulator for Clifford circuits based on a SAT encoding. The authors also discuss a SAT encoding for simulating universal circuits [17, 189] of exponential size, making it impractical. Subsequently, [107] achieved the strong simulation of universal quantum circuits by a linear encoding as a (weighted) model counting problem. The crux of the method is a generalization of the stabilizer formalism to universal quantum computing which effectively discretizes the state space. Empirical evaluation shows that the method is competitive to state-of-the-art methods based on DD and ZX calculus (see Section 6).

SAT-based solvers have also been used for the reversible simulation of irreversible (i.e. classical) circuits [190, 110], which is relevant for the construction of space-efficient quantum oracles, e.g. for Grover's algorithm [71] and quantum backtracking algorithms [114, 142]. This was later improved by the spooky pebble game [96], which is a model to trade off classical and quantum space. Using SAT solvers, the spooky pebble game has been used to study trade-offs between classical space, quantum space, and circuit depth of a computation, optimizing quantum circuits within hardware constraints [139, 140]. SAT has also been used for equivalence checking of Clifford circuits, using the stabilizer formalism [17], though the problem is in P [165]. For universal quantum circuits, equivalence checking can be achieved via a Turing reduction to model counting, as proved by [108] based on the simulation through model counting approach mentioned above [107] and an equivalence checking approach [165].

Clifford circuit synthesis deals with the problem of finding a Clifford circuit, which, from the $|0...0\rangle$ state, generates a particular target stabilizer state (a form of state preparation). SAT-based methods have been applied to this problem, both with additional optimality constraints [148, 57, 102, 116, 153, 55, 125] and without [17]. The work [148] uses both a MaxSAT solver [21] and a regular SAT solver in combination with binary search. Additional constraints (such as searching for the shortest circuit) are important because without these, the problem is known to be in P [1]. And while SAT solvers are good at hard problems, they are impractical for tractable problems, as demonstrated in [165].

A different application of SAT solvers has been in optimal layout synthesis, where swap operations are inserted into the circuit to ensure that multi-qubit

gates are only executed on connected physical qubits. Minimizing the number of swaps is NP-hard [105] and thus the problem was tackled with SMT [161, 27, 75] and MaxSAT-based methods [112].

Related is the synthesis of circuits that transform one "graph state" into another using only single-qubit operations. Graph states are a special type of stabilizer states that play a crucial role in quantum networking as well as measurement-based quantum computing [141, 78]. An n-qubit graph state is a quantum state that can be described by an undirected graph $G = (V, E)$ with n vertices V and edges $E \subseteq V \times V$. Intuitively, the edges of the graph capture information about the way the qubits (corresponding to vertices in the graph) are entangled. Formally, a graph state $|G\rangle$ is the state $(\prod_{(v,w)\in E} CZ_{v,w})H^{\otimes n}\,|0...0\rangle$, where $CZ_{v,w}$ is a two qubit quantum gate equal to $H_w \mathrm{CNOT}_{v,w} H_w$.

An important property of graph states is that several important operations, i.e. single- and two-qubit Clifford gates and single-qubit standard measurements, can be expressed in terms of graph transformations. This allows for reasoning about transformations of graph states entirely in terms of transformations of graphs. A recurring problem [50, 77, 87] is finding transformations between graph states using only local (i.e. single-qubit) operations, a problem which is NP-complete in general [49]. By reducing the problem to SAT, this problem has been successfully solved up to 17 qubits [26].

6 Graphical Calculus-based Methods

Another notable technique for quantum circuit compilation is the use of graphical calculi. These techniques consist of using a set of composable graphical generators that interact and can be rewritten via a set of equations. In recent years there have been many proposals for representing quantum systems as graphical languages. For example, there exist graphical languages with a focus on linear optical circuits [42, 79, 64], fermionic quantum systems [54], open quantum systems [194], Gaussian pure states [109], qudits [101], and more general systems [44, 18]. Even more so, there has recently been a diagrammatic axiomatization of quantum circuits [41]. Still, the most prominent graphical language for quantum circuit compilation is the *ZX-calculus* [43], (the focus of this section) and its multiple variations and extensions [7, 76, 136, 181, 173].

Initially developed by Coecke and Duncan [43], the ZX-calculus is a graphical language used for reasoning about quantum systems and quantum computations. Inspired by Penrose's tensor network notation [134] and graphical languages stemming from category theory [150], ZX-diagrams consist of a set of graphical generators that semantically represent linear maps $f\colon \mathbb{C}^{2^n} \to \mathbb{C}^{2^m}$ between Hilbert spaces. We can compose these generators both sequentially and parallelly (corresponding to multiplication and Kronecker product of the linear maps they represent, respectively) to create more complex quantum systems. In fact, ZX-diagrams are *universal* in the sense that they can represent any linear map between qubit Hilbert spaces, not necessarily limited to quantum operations (and interestingly, it was shown in [175] how to do a conversion between QMDDs

and ZH-diagrams, which in turn can be converted into ZX-diagrams). This makes ZX-diagrams more flexible than standard circuit notation. ZX-diagrams are also equipped with a set of rewrite rules, forming the ZX-calculus. These rewrite rules identify semantical identities between diagrams, and they are the basis for ZX-calculus-based algorithms, for which we are going to give an overview now. Figure 4 shows an example correspondence between a quantum circuit and equivalent ZX-diagram (up to scalar factor).

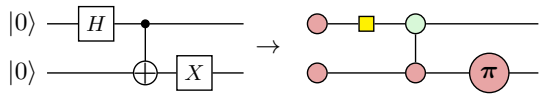


Fig. 4. A quantum circuit and corresponding ZX-diagram.

When it comes to simulation of quantum circuits with ZX-calculus, most approaches [92, 95, 93, 35] opt for calculating individual amplitudes of the state vector using variations of the following set of steps. We start with a quantum circuit consisting of an input state, a sequence of gates, and an effect $\langle x|$ that we turn into a ZX-diagram. By applying some specific rewrite rules, we turn our diagram into a *graph-like* ZX-diagram [59]. We then contract the resulting diagram via a specific set of rewrite rules that strictly decrease the number of nodes (to ensure termination), and then split the reduced diagram into smaller ones by identifying sub-diagrams where we can apply a stabilizer decomposition. This results in a linear combination of ZX-diagrams that can be recursively simplified by again contracting and decomposing each diagram, resulting in a linear combination of smaller ZX-diagrams that we then semantically translate to calculate the resulting amplitude. Adjacent to the simulation of quantum circuits, recent techniques allow for differentiation and integration of ZX-diagrams with applications in quantum machine learning [167, 182, 85].

When the task is optimizing a circuit with ZX-calculus [158, 13, 81, 91, 90], we also proceed by translating the input circuit into a ZX-diagram, then to a graph-like diagram, followed by similar simplification rules until termination, and then extracting a circuit from the resulting simplified diagram. Since ZX-diagrams are more general than quantum circuits, the process of extracting a circuit from a unitary ZX-diagram is #P-hard [14]. Fortunately, there are sufficient conditions that have been identified for circuit extraction to be done efficiently [8, 59, 185, 60]. These are graph-theoretical *flow* conditions on the diagrams ensuring that a circuit can be extracted step-wise from them. The aforementioned circuit optimization technique can be augmented for quantum-classical circuits (i.e. quantum circuits that can interact with classical circuits) [23] for an extension of the ZX-calculus that allows the discarding of quantum systems [36].

When it comes to circuit synthesis (and also very much related to circuit optimization), most works revolve around *Pauli exponentials* [47, 185]. Exponentiated Paulis are operations of the form $e^{-i\alpha P_1 \otimes P_2 \otimes \ldots \otimes P_n}$ and arise naturally when performing quantum chemistry simulations [47]. These operations have a

compact representation as ZX-diagrams (referred to as *Pauli gadgets*) and their interactions (e.g. commutation rules between themselves and other quantum gates) can be intuitively reasoned about as ZX-diagram rewrite rules. Given a computation in the form of a composition of Pauli exponentials, the task of ZX-calculus based synthesis aims for a succinct representation of these operations in the form of a sequence of gates (that is, a circuit-like ZX-diagram), oftentimes taking into account device constraints such as qubit connectivity when choosing the layout of multi-qubit gates [192, 48, 70, 68].

Lastly, we want to mention the task of circuit equivalence checking using ZX-calculus. As a language, the ZX-calculus is *complete* in the sense that there are sets of rewrite rules [174] that can provably transform any two semantically equivalent ZX-diagrams into one another. This does not mean that the procedure to do so (e.g., via normal forms) is optimal, so for equivalence a different approach [91, 131, 90, 133] exists: Given two quantum circuits U, U' (in the form of ZX-diagrams), composing one with the dagger of the other in order to verify $U^\dagger U' = I$ (up to global phase). This would mean that after reducing the composed circuits we should arrive to the identity diagram, which is a collection of bare wires. The simplification algorithm [91] also boils down to turning a diagram to graph-like form and applying strictly reducing rewrites until no more nodes can be simplified. It was shown in [131] that this technique does not arrive to the identity diagram on certain (desirable) cases, such as when the two circuits differ by small errors or when one of them uses an ancillary qubit.

7 Other Quantum Circuit Compilation Methods

Without the intention to be complete, we present other important automated tools used in quantum circuit compilation in this section.

Bisimulation has been used in computer science for inter alia process analysis [146] but can also be used in quantum simulation. This line of research was introduced by Jimenez et al. [86], showing that this technique complements quantum simulation methods with decision diagrams.

Shaik and van der Pol [151] tackled circuit topology optimization with modern planning tools. Interestingly, the resulting optimizer is shown to outperform methods based on satisfiability. Venturelli et al. [168] also use planning and constraint programming to realize a quantum circuit optimization tool.

Quantum circuit optimization problems have also been solved with (mixed integer) linear programming. The tackled problems include finding optimal quantum circuits under restrictions on the topology (qubit connectivity) [119, 180] or under restrictions on the gate set [117, 11].

Finally, tensor networks are used in physics for simulation of physical quantum systems [194, 127] and have found various applications in quantum computing [172, 129, 149]. Tensor networks represent quantum information and operations in similar way as graph calculi (see Section 6). In fact, ZX-diagrams are tensor networks [185], but with an additional set of rewrite rules to find simplification strategies or to carry out proofs in purely diagrammatic form. A

matrix product states (MPS) [135] is a linear tensor network with a variable order much like in decision diagrams (see Section 4). In fact, Vinkhuijzen et al. [177] found that MPS can polynomially simulate EVDD (QMDD), but not vice versa. Like quantum circuits [61, 15], tensor networks have also been used in machine learning [143]. A historical overview can be found in [126].

Finally, we point out that tensor networks are used in similar ways as decision diagrams and satisfiability. In formal verification, for instance, sets of states and transitions are represented in DD, before fixpoints are computed using matrix-vector operations [106, 66, 25, 56]. Tensor networks also treat single quantum states and operations that way (see MPS and MPO [127]). Similarly, the SAT-based approaches identified in Section 5 essentially use a bounded model checking [20] encoding of circuits. Moreover, in knowledge compilation in AI, information is first compiled in a succinct representation before analyzing it using queries, which is how tensor networks are also used in physics [126].

8 Conclusion

This survey found various types of decision diagram-, SAT- and graph-calculus-based approaches are used extensively in quantum circuit compilation. The result illustrates how quantum circuit compilation benefited from automated reasoning techniques originally developed for reasoning about classical systems. Perhaps surprising, is that not only universal quantum simulation has been efficiently handled with decision diagrams, model counting and ZX-calculus-based approaches, but also harder problems like (universal) equivalence checking. In fact, together these methods make up the state-of-the-art in quantum circuit simulation, often with different performance characteristics (e.g., model counting can be slower than decision diagrams on structured circuits but better on more unstructured circuits, while ZX-calculus can sometimes outperform both methods). Similarly, optimization and synthesis tasks have been shown to be solvable with DD, SAT/SMT and ZX-calculus for non-universal cases (e.g., for Clifford circuits) and for universal circuits, which are however mostly handled with heuristic approaches. In the latter case, ZX-calculus seems to excel.

From these successes, and recent progress in this direction [144, 80], we conclude that 'classical' automated reasoning tools will likely come to play a larger role in other quantum computing and physics applications, such as finding ground states, phase transitions and quantum error correction, where currently methods like tensor networks are often applied.

Acknowledgments

This work was supported by the Dutch National Growth Fund, as part of the Quantum Delta NL program. This work was funded by the European Union under the NEASQC project (Grant Agreement No. 951821) and the EQUALITY project (Grant Agreement No. 101080142). This publication is part of the project Divide & Quantum (with project number 1389.20.241) of the research program NWA-ORC which is (partly) financed by the Dutch Research Council (NWO).

References

1. Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Physical Review A 70(5) (nov 2004), https://doi.org/10.1103%2Fphysreva.70.052328
2. Abdollahi, A., Pedram, M.: Analysis and synthesis of quantum circuits by using quantum decision diagrams. In: Proceedings of the Design Automation & Test in Europe Conference. vol. 1, pp. 1–6. IEEE (2006)
3. Akers: Binary decision diagrams. IEEE Transactions on Computers C-27(6), 509–516 (1978)
4. Amy, M.: Towards large-scale functional verification of universal quantum circuits. arXiv:1805.06908 (2018)
5. Amy, M.: Formal methods in quantum circuit design (PhD thesis). Ph.D. thesis, University of Waterloo (2019)
6. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A., et al.: Quantum supremacy using a programmable superconducting processor. Nature 574(7779), 505–510 (2019)
7. Backens, M., Kissinger, A.: ZH: A complete graphical calculus for quantum computations involving classical non-linearity. Electronic Proceedings in Theoretical Computer Science 287, 23–42 (Jan 2019), http://dx.doi.org/10.4204/EPTCS.287.2
8. Backens, M., Miller-Bakewell, H., de Felice, G., Lobski, L., van de Wetering, J.: There and back again: A circuit extraction tale. Quantum 5, 421 (Mar 2021), http://dx.doi.org/10.22331/q-2021-03-25-421
9. Bahar, R.I., Frohm, E.A., Gaona, C.M., Hachtel, G.D., Macii, E., Pardo, A., Somenzi, F.: Algebraic decision diagrams and their applications. Formal methods in system design 10(2-3), 171–206 (1997)
10. Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T.C., Mutus, J., Fowler, A.G., Campbell, B., Chen, Y., Chen, Z., Chiaro, B., Dunsworth, A., Neill, C., O'Malley, P., Roushan, P., Vainsencher, A., Wenner, J., Korotkov, A.N., Cleland, A.N., Martinis, J.M.: Superconducting quantum circuits at the surface code threshold for fault tolerance. Nature 508(7497), 500–503 (Apr 2014), https://doi.org/10.1038/nature13171
11. Baßler, P., Zipper, M., Cedzich, C., Heinrich, M., Huber, P.H., Johanning, M., Kliesch, M.: Synthesis of and compilation with time-optimal multi-qubit gates. Quantum 7, 984 (2023), https://doi.org/10.22331/q-2023-04-20-984
12. Bauer-Marquart, F., Leue, S., Schilling, C.: symQV: Automated symbolic verification of quantum programs. In: Formal Methods: 25th International Symposium, FM 2023, Lübeck, Germany, March 6–10, 2023, Proceedings. pp. 181–198. Springer (2023)
13. de Beaudrap, N., Bian, X., Wang, Q.: Fast and Effective Techniques for T-Count Reduction via Spider Nest Identities. In: Flammia, S.T. (ed.) 15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020). Leibniz International Proceedings in Informatics (LIPIcs), vol. 158, pp. 11:1–11:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2020), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2020.11
14. de Beaudrap, N., Kissinger, A., van de Wetering, J.: Circuit extraction for ZX-diagrams can be #P-hard. In: Bojańczyk, M., Merelli, E., Woodruff, D.P. (eds.) 49th International Colloquium on Automata, Languages, and Programming (ICALP 2022). Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2022), https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2022.119

15. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quantum Science and Technology 4(4), 043001 (nov 2019), https://dx.doi.org/10.1088/2058-9565/ab4eb5
16. Berent, L., Burgholzer, L., Derks, P.J.H., Eisert, J., Wille, R.: Decoding quantum color codes with MaxSAT. arXiv preprint arXiv:2303.14237 (2023), https://doi.org/10.48550/arXiv.2303.14237
17. Berent, L., Burgholzer, L., Wille, R.: Towards a SAT Encoding for Quantum Circuits: A Journey From Classical Circuits to Clifford Circuits and Beyond. In: Meel, K.S., Strichman, O. (eds.) 25th International Conference on Theory and Applications of Satisfiability Testing (SAT 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 236, pp. 18:1–18:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.SAT.2022.18
18. Bergholm, V., Biamonte, J.D.: Categorical quantum circuits. Journal of Physics A: Mathematical and Theoretical 44(24), 245304 (2011)
19. Bertot, Y., Castéran, P.: Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions. Springer Science & Business Media (2013)
20. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded model checking. Handbook of satisfiability 185(99), 457–481 (2009)
21. Biere, A., Heule, M., van Maaren, H.: Handbook of satisfiability, vol. 185. IOS press (2009)
22. Boole, G.: An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities. Walton and Maberly (1854)
23. Borgna, A., Perdrix, S., Valiron, B.: Hybrid quantum-classical circuit simplification with the zx-calculus. In: Oh, H. (ed.) Programming Languages and Systems. pp. 121–139. Springer International Publishing, Cham (2021)
24. Bove, A., Dybjer, P., Norell, U.: A brief overview of Agda – a functional language with dependent types. In: Theorem Proving in Higher Order Logics: 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings 22. pp. 73–78. Springer (2009)
25. Brand, S., Bäck, T., Laarman, A.: A decision diagram operation for reachability. In: International Symposium on Formal Methods. pp. 514–532. Springer (2023)
26. Brand, S., Coopmans, T., Laarman, A.: Quantum graph-state synthesis with SAT. Proceedings of the 14th International Workshop on Pragmatics of SAT (2023)
27. Brandhofer, S., Kim, J., Niu, S., Bronn, N.T.: SAT-based quantum circuit adaptation. In: 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1–6. IEEE (2023)
28. Bravyi, S., Browne, D., Calpin, P., Campbell, E., Gosset, D., Howard, M.: Simulation of quantum circuits by low-rank stabilizer decompositions. Quantum 3, 181 (Sep 2019), https://doi.org/10.22331/q-2019-09-02-181
29. Bryant, R.E.: Symbolic Boolean manipulation with ordered binary-decision diagrams. ACM Comput. Surv. 24(3), 293–318 (sep 1992)
30. Burgholzer, L., Bauer, H., Wille, R.: Hybrid Schrödinger-Feynman simulation of quantum circuits with decision diagrams. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). pp. 199–206 (2021)
31. Burgholzer, L., Kueng, R., Wille, R.: Random stimuli generation for the verification of quantum circuits. In: Proceedings of the 26th Asia and South Pacific Design Automation Conference. pp. 767–772 (2021)

32. Burgholzer, L., Wille, R.: Advanced equivalence checking for quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 40(9), 1810–1824 (2020)
33. Burgholzer, L., Wille, R.: Improved DD-based equivalence checking of quantum circuits. In: 25th Asia and South Pacific Design Automation Conference (ASP-DAC). pp. 127–132 (2020)
34. Burgholzer, L., Wille, R.: QCEC: A JKQ tool for quantum circuit equivalence checking. Software Impacts 7, 100051 (2021)
35. Cam, T., Martiel, S.: Speeding up quantum circuits simulation using ZX-calculus. arXiv preprint arXiv:2305.02669 (2023)
36. Carette, T., Jeandel, E., Perdrix, S., Vilmart, R.: Completeness of graphical languages for mixed state quantum mechanics. ACM Transactions on Quantum Computing 2(4) (dec 2021), https://doi.org/10.1145/3464693
37. Chakraborty, S., Fremont, D., Meel, K., Seshia, S., Vardi, M.: Distribution-aware sampling and weighted model counting for sat. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 28 (2014)
38. Chareton, C., Bardin, S., Lee, D., Valiron, B., Vilmart, R., Xu, Z.: Formal methods for quantum programs: A survey. arXiv preprint arXiv:2109.06493 (2021)
39. Chen, Y.F., Rümmer, P., Tsai, W.L.: A theory of cartesian arrays (with applications in quantum circuit verification). In: International Conference on Automated Deduction. pp. 170–189. Springer (2023)
40. Clarke, E.M., McMillan, K.L., Zhao, X., Fujita, M., Yang, J.: Spectral transforms for large Boolean functions with applications to technology mapping. In: Proceedings of the 30th international Design Automation Conference. pp. 54–60 (1993)
41. Clément, A., Delorme, N., Perdrix, S., Vilmart, R.: Quantum circuit completeness: Extensions and simplifications. In: Murano, A., Silva, A. (eds.) 32nd EACSL Annual Conference on Computer Science Logic (CSL 2024). Leibniz International Proceedings in Informatics (LIPIcs), vol. 288, pp. 20:1–20:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2024), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.CSL.2024.20
42. Clément, A., Heurtel, N., Mansfield, S., Perdrix, S., Valiron, B.: LOv-Calculus: A Graphical Language for Linear Optical Quantum Circuits. In: Szeider, S., Ganian, R., Silva, A. (eds.) 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 241, pp. 35:1–35:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2022.35
43. Coecke, B., Duncan, R.: Interacting Quantum Observables: Categorical Algebra and Diagrammatics. New Journal of Physics 13(4), 043016 (Apr 2011), http://arxiv.org/abs/0906.4725, arXiv:0906.4725 [quant-ph]
44. Coecke, B., Kissinger, A.: Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning. Cambridge University Press (2017)
45. Cook, S.A.: The complexity of theorem-proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing. p. 151–158. STOC '71, Association for Computing Machinery, New York, NY, USA (1971), https://doi.org/10.1145/800157.805047
46. Córcoles, A.D., Kandala, A., Javadi-Abhari, A., McClure, D.T., Cross, A.W., Temme, K., Nation, P.D., Steffen, M., Gambetta, J.M.: Challenges and opportunities of near-term quantum computing systems. arXiv:1910.02894 (2019)

47. Cowtan, A., Dilkes, S., Duncan, R., Simmons, W., Sivarajah, S.: Phase gadget synthesis for shallow circuits. Electronic Proceedings in Theoretical Computer Science 318, 213–228 (May 2020), http://dx.doi.org/10.4204/EPTCS.318.13
48. Cowtan, A., Simmons, W., Duncan, R.: A generic compilation strategy for the unitary coupled cluster ansatz. arXiv preprint arXiv:2007.10515 (2020), https://doi.org/10.48550/arXiv.2007.10515
49. Dahlberg, A., Helsen, J., Wehner, S.: How to transform graph states using single-qubit operations: computational complexity and algorithms. Quantum Science and Technology 5(4), 045016 (2020)
50. Dahlberg, A., Wehner, S.: Transforming graph states using single-qubit operations. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376(2123), 20170325 (2018)
51. Darwiche, A.: SDD: A new canonical representation of propositional knowledge bases. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
52. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
53. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM 5(7), 394–397 (1962)
54. De Felice, G., Hadzihasanovic, A., Ng, K.F.: A diagrammatic calculus of fermionic quantum circuits. Logical Methods in Computer Science 15 (2019)
55. Deng, H., Tao, R., Peng, Y., Wu, X.: A case for synthesis of recursive quantum unitary programs. Proceedings of the ACM on Programming Languages 8(POPL), 1759–1788 (2024)
56. van Dijk, T., Laarman, A., van de Pol, J.: Multi-core BDD operations for symbolic reachability. ENTCS 296, 127–143 (2013)
57. Ding, J., Yamashita, S.: Exact synthesis of nearest neighbor compliant quantum circuits in 2-D architecture and its application to large-scale circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39(5), 1045–1058 (2019)
58. Ding, Y., Chong, F.T.: Circuit synthesis and compilation. In: Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers, pp. 91–125. Springer (2020)
59. Duncan, R., Kissinger, A., Perdrix, S., van de Wetering, J.: Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. Quantum 4, 279 (Jun 2020), https://doi.org/10.22331/q-2020-06-04-279
60. Duncan, R., Perdrix, S.: Rewriting measurement-based quantum computations with generalised flow. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) Automata, Languages and Programming. pp. 285–296. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
61. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain (2017)
62. Fargier, H., Marquis, P., Schmidt, N.: Semiring labelled decision diagrams, revisited: Canonicity and spatial efficiency issues. In: IJCAI. pp. 884–890 (2013)
63. Feigenbaum, J., Kannan, S., Vardi, M.Y., Viswanathan, M.: Complexity of problems on graphs represented as OBDDs. In: STACS. pp. 216–226. Springer (1998)
64. de Felice, G., Coecke, B.: Quantum linear optics via string diagrams. Electronic Proceedings in Theoretical Computer Science 394, 83–100 (Nov 2023), http://dx.doi.org/10.4204/EPTCS.394.6
65. Finigan, W., Cubeddu, M., Lively, T., Flick, J., Narang, P.: Qubit allocation for noisy intermediate-scale quantum computers. arXiv:1810.08291 (2018)

66. Fujita, M., McGeer, P.C., Yang, J.Y.: Multi-terminal binary decision diagrams: An efficient data structure for matrix representation. FMSD 10(2-3), 149–169 (1997)
67. Giles, B., Selinger, P.: Exact synthesis of multiqubit Clifford+T circuits. Physical Review A 87(3), 032332 (2013)
68. Gogioso, S., Yeung, R.: Annealing optimisation of mixed ZX phase circuits. Electronic Proceedings in Theoretical Computer Science 394, 415–431 (Nov 2023), http://dx.doi.org/10.4204/EPTCS.394.20
69. Gottesman, D.: Stabilizer codes and quantum error correction. arXiv:quant-ph/9705052 (1997)
70. Meijer-van de Griend, A., Duncan, R.: Architecture-aware synthesis of phase polynomials for NISQ devices. Electronic Proceedings in Theoretical Computer Science 394, 116–140 (Nov 2023), http://dx.doi.org/10.4204/EPTCS.394.8
71. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
72. Grurl, T., Fuß, J., Wille, R.: Considering decoherence errors in the simulation of quantum circuits using decision diagrams. In: Proceedings of the 39th International Conference on Computer-Aided Design. pp. 1–7 (2020)
73. Grurl, T., Fuß, J., Wille, R.: Noise-aware quantum circuit simulation with decision diagrams. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 42(3), 860–873 (2022)
74. Grurl, T., Kueng, R., Fuß, J., Wille, R.: Stochastic quantum circuit simulation using decision diagrams. In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 194–199. IEEE (2021)
75. Guo, Z.H., Wang, T.C.: SMT-based layout synthesis approaches for quantum circuits. In: Proceedings of the 2024 International Symposium on Physical Design. p. 235–243. ISPD '24, Association for Computing Machinery, New York, NY, USA (2024), https://doi.org/10.1145/3626184.3633316
76. Hadzihasanovic, A.: A diagrammatic axiomatisation for qubit entanglement (2015)
77. Hahn, F., Pappa, A., Eisert, J.: Quantum network routing and local complementation. npj Quantum Information 5(1), 76 (2019)
78. Hein, M., Dür, W., Eisert, J., Raussendorf, R., Nest, M., Briegel, H.J.: Entanglement in graph states and its applications. arXiv:0602096 (2006)
79. Heurtel, N.: A complete graphical language for linear optical circuits with finite-photon-number sources and detectors (2024)
80. Hillmich, S., Hadfield, C., Raymond, R., Mezzacapo, A., Wille, R.: Decision diagrams for quantum measurements with shallow circuits. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE). pp. 24–34. IEEE (2021)
81. Holker, C.: Causal flow preserving optimisation of quantum circuits in the ZX-calculus. arXiv preprint arXiv:2312.02793 (2023), https://doi.org/10.48550/arXiv.2312.02793
82. Hong, X., Ying, M., Feng, Y., Zhou, X., Li, S.: Approximate equivalence checking of noisy quantum circuits. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 637–642 (2021)
83. Hong, X., Zhou, X., Li, S., Feng, Y., Ying, M.: A tensor network based decision diagram for representation of quantum circuits. ACM Transactions on Design Automation of Electronic Systems (TODAES) 27(6), 1–30 (2022)

84. Janzing, D., Wocjan, P., Beth, T.: " non-identity-check" is QMA-complete. International Journal of Quantum Information 3(03), 463–473 (2005)
85. Jeandel, E., Perdrix, S., Veshchezerova, M.: Addition and Differentiation of ZX-Diagrams. In: Felty, A.P. (ed.) 7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 228, pp. 13:1–13:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.FSCD.2022.13
86. Jiménez-Pastor, A., Larsen, K.G., Tribastone, M., Tschaikowski, M.: Efficient simulation of quantum circuits by model order reduction. arXiv preprint arXiv:2308.09510 (2023)
87. de Jong, J., Hahn, F., Tcholtchev, N., Hauswirth, M., Pappa, A.: Extracting maximal entanglement from linear cluster states. arXiv preprint arXiv:2211.16758 (2022)
88. Jozsa, R., van den Nest, M.: Classical simulation complexity of extended Clifford circuits. Quantum Inf. Comput. 14(7-8), 633–648 (2014), https://doi.org/10.26421/QIC14.7-8-7
89. Kim, Y., Eddins, A., Anand, S., Wei, K.X., Van Den Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K., et al.: Evidence for the utility of quantum computing before fault tolerance. Nature 618(7965), 500–505 (2023)
90. Kissinger, A., van de Wetering, J.: PyZX: Large scale automated diagrammatic reasoning. In: QPL (2019), https://api.semanticscholar.org/CorpusID:104292461
91. Kissinger, A., van de Wetering, J.: Reducing the number of non-Clifford gates in quantum circuits. Physical Review A 102(2) (Aug 2020), http://dx.doi.org/10.1103/PhysRevA.102.022406
92. Kissinger, A., van de Wetering, J.: Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions. Quantum Science and Technology 7(4), 044001 (Oct 2022), http://arxiv.org/abs/2109.01076, arXiv:2109.01076 [quant-ph]
93. Kissinger, A., van de Wetering, J., Vilmart, R.: Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions. In: Le Gall, F., Morimae, T. (eds.) 17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 232, pp. 5:1–5:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2022.5
94. Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and quantum computation. American Mathematical Soc. (2002)
95. Koch, M., Yeung, R., Wang, Q.: Speedy contraction of ZX diagrams with triangles via stabiliser decompositions. arXiv preprint arXiv:2307.01803 (2023), https://doi.org/10.48550/arXiv.2307.01803
96. Kornerup, N., Sadun, J., Soloveichik, D.: The spooky pebble game. arXiv preprint arXiv:2110.08973 (2021), https://doi.org/10.48550/arXiv.2110.08973
97. Lai, Y.T., Pedram, M., Vrudhula, S.B.: EVBDD-based algorithms for integer linear programming, spectral transformation, and function decomposition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 13(8), 959–975 (1994)
98. Landahl, A.J., Anderson, J.T., Rice, P.R.: Fault-tolerant quantum computing with color codes. arXiv preprint arXiv:1108.5738 (2011), https://doi.org/10.48550/arXiv.1108.5738

99. Leino, K.R.M.: Dafny: An automatic program verifier for functional correctness. In: International conference on logic for programming artificial intelligence and reasoning. pp. 348–370. Springer (2010)
100. Lewis, M., Soudjani, S., Zuliani, P.: Formal verification of quantum programs: Theory, tools, and challenges. ACM Transactions on Quantum Computing 5(1), 1–35 (2023)
101. Lin, R.: A graphical calculus for quantum computing with multiple qudits using generalized Clifford algebras (2023)
102. Lin, S.W., Chen, S.H., Wang, T.F., Chen, Y.R.: A quantum SMT solver for bit-vector theory. arXiv preprint arXiv:2303.09353 (2023)
103. Linden, N., de Wolf, R.: Lightweight detection of a small number of large errors in a quantum circuit. Quantum 5, 436 (2021)
104. Lloyd, S.: Universal quantum simulators. Science 273(5278), 1073–1078 (1996), https://www.science.org/doi/abs/10.1126/science.273.5278.1073
105. Maslov, D., Falconer, S.M., Mosca, M.: Quantum circuit placement. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 27(4), 752–763 (2008)
106. McMillan, K.L.: Symbolic model checking: An approach to the state explosion problem. Ph.D. thesis, Carnegie Mellon University (1992)
107. Mei, J., Bonsangue, M., Laarman, A.: Simulating quantum circuits by model counting. arXiv preprint arXiv:2403.07197 (2024)
108. Mei, J., Coopmans, T., Bonsangue, M., Laarman, A.: Equivalence checking of quantum circuits by model counting. arXiv preprint (to appear) (2024)
109. Menicucci, N.C., Flammia, S.T., van Loock, P.: Graphical calculus for Gaussian pure states. Physical Review A 83(4), 042335 (2011)
110. Meuli, G., Soeken, M., De Micheli, G.: SAT-based {CNOT, T} quantum circuit synthesis. In: Kari, J., Ulidowski, I. (eds.) Reversible Computation. pp. 175–188. Springer International Publishing, Cham (2018)
111. Miller, D.M., Thornton, M.A.: QMDD: A decision diagram structure for reversible and quantum circuits. 36th International Symposium on Multiple-Valued Logic (ISMVL'06) pp. 30–30 (2006)
112. Molavi, A., Xu, A., Diges, M., Pick, L., Tannu, S., Albarghouthi, A.: Qubit mapping and routing via MaxSAT. In: 2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO). pp. 1078–1091. IEEE (2022)
113. Montanaro, A.: Quantum algorithms: an overview. npj Quantum Information 2(1), 15023 (Jan 2016), https://doi.org/10.1038/npjqi.2015.23
114. Montanaro, A.: Quantum-walk speedup of backtracking algorithms. Theory of Computing 14(1), 1–24 (2018)
115. de Moura, L., Kong, S., Avigad, J., Van Doorn, F., von Raumer, J.: The Lean theorem prover (system description). In: Automated Deduction-CADE-25: 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings 25. pp. 378–388. Springer (2015)
116. Murali, P., Javadi-Abhari, A., Chong, F.T., Martonosi, M.: Formal constraint-based compilation for noisy intermediate-scale quantum systems. Microprocessors and Microsystems 66, 102–112 (2019)
117. Nagarajan, H., Lockwood, O., Coffrin, C.: QuantumCircuitOpt: An open-source framework for provably optimal quantum circuit design. In: 2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS). pp. 55–63. IEEE (2021)

118. Nakamura, K., Denzumi, S., Nishino, M.: Variable Shift SDD: A More Succinct Sentential Decision Diagram. In: Faro, S., Cantone, D. (eds.) 18th International Symposium on Experimental Algorithms (SEA 2020). Leibniz International Proceedings in Informatics (LIPIcs), vol. 160, pp. 22:1–22:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2020), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.SEA.2020.22
119. Nannicini, G., Bishop, L.S., Günlük, O., Jurcevic, P.: Optimal qubit assignment and routing via integer programming. ACM Transactions on Quantum Computing 4(1), 1–31 (2022)
120. van den Nest, M.: Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. Quantum Information & Computation 10(3), 258–271 (2010)
121. Nielsen, M.A., Chuang, I.L.: Quantum information and quantum computation. Cambridge: Cambridge University Press 2(8), 23 (2000)
122. Niemann, P., Wille, R., Drechsler, R.: Equivalence checking in multi-level quantum systems. In: Reversible Computation: 6th International Conference, RC 2014, Kyoto, Japan, July 10-11, 2014. Proceedings 6. pp. 201–215. Springer (2014)
123. Niemann, P., Wille, R., Drechsler, R.: Advanced exact synthesis of Clifford+T circuits. Quantum Information Processing 19, 1–23 (2020)
124. Niemann, P., Zulehner, A., Drechsler, R., Wille, R.: Overcoming the tradeoff between accuracy and compactness in decision diagrams for quantum computation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 39(12), 4657–4668 (2020)
125. Oliveira Oliveira, M.d.: On the satisfiability of quantum circuits of small treewidth. Theory of Computing Systems 61, 656–688 (2017)
126. Orús, R.: Tensor networks for complex quantum systems. Nature Reviews Physics 1(9), 538–550 (2019)
127. Orús, R.: A practical introduction to tensor networks: Matrix product states and projected entangled pair states. Annals of Physics 349, 117–158 (2014), https://www.sciencedirect.com/science/article/pii/S0003491614001596
128. Oztok, U., Darwiche, A.: A top-down compiler for sentential decision diagrams. In: IJCAI. p. 3141–3148. IJCAI'15, AAAI Press (2015)
129. Pan, F., Zhang, P.: Simulation of quantum circuits using the big-batch tensor network method. Physical Review Letters 128(3), 030501 (2022)
130. Paulson, L.C.: Isabelle: A generic theorem prover. Springer (1994)
131. Peham, T., Burgholzer, L., Wille, R.: Equivalence checking of quantum circuits with the ZX-calculus. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 12(3), 662–675 (2022)
132. Peham, T., Burgholzer, L., Wille, R.: Equivalence checking of quantum circuits with the zx-calculus. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 12(3), 662–675 (2022)
133. Peham, T., Burgholzer, L., Wille, R.: Equivalence checking of parameterized quantum circuits: Verifying the compilation of variational quantum algorithms. In: 2023 28th Asia and South Pacific Design Automation Conference (ASP-DAC). pp. 702–708 (2023)
134. Penrose, R.: Applications of negative dimensional tensors. In: Combinatorial Mathematics and its Applications. Academic Press (1971)
135. Perez-Garcia, D., Verstraete, F., Wolf, M., Cirac, J.: Matrix product state representations. Quantum Information & Computation 7(5), 401–430 (2007), https://dl.acm.org/doi/10.5555/2011832.2011833

136. Poór, B., Wang, Q., Shaikh, R.A., Yeh, L., Yeung, R., Coecke, B.: Completeness for arbitrary finite dimensions of zxw-calculus, a unifying calculus. In: 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). IEEE (Jun 2023), http://dx.doi.org/10.1109/LICS56636.2023.10175672
137. Preskill, J.: Quantum computing and the entanglement frontier. Bulletin of the American Physical Society 58 (2013)
138. Preskill, J.: Quantum Computing in the NISQ era and beyond. Quantum 2, 79 (Aug 2018), https://doi.org/10.22331/q-2018-08-06-79
139. Quist, A.J., Laarman, A.: Optimizing quantum space using spooky pebble games. In: International Conference on Reversible Computation. pp. 134–149. Springer (2023)
140. Quist, A.J., Laarman, A.: Trade-offs between classical and quantum space using spooky pebbling. arXiv preprint arXiv:2401.10579 (2024)
141. Raussendorf, R., Briegel, H.J.: A one-way quantum computer. Phys. Rev. Lett. 86, 5188–5191 (May 2001), https://link.aps.org/doi/10.1103/PhysRevLett.86.5188
142. Rennela, M., Brand, S., Laarman, A., Dunjko, V.: Hybrid divide-and-conquer approach for tree search algorithms. Quantum 7, 959 (2023)
143. Rieser, H.M., Köster, F., Raulf, A.P.: Tensor networks for quantum machine learning. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 479(2275), 20230218 (2023)
144. Sander, A., Burgholzer, L., Wille, R.: Towards hamiltonian simulation with decision diagrams. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). vol. 1, pp. 283–294. IEEE (2023)
145. Sang, T., Bacchus, F., Beame, P., Kautz, H.A., Pitassi, T.: Combining component caching and clause learning for effective model counting. In: International Conference on Theory and Applications of Satisfiability Testing (2004), https://api.semanticscholar.org/CorpusID:52027
146. Sangiorgi, D.: Introduction to bisimulation and coinduction. Cambridge University Press (2011)
147. Sanner, S., McAllester, D.: Affine algebraic decision diagrams (AADDs) and their application to structured probabilistic inference. In: IJCAI. vol. 2005, pp. 1384–1390 (2005), https://dl.acm.org/doi/abs/10.5555/1642293.1642513
148. Schneider, S., Burgholzer, L., Wille, R.: A SAT encoding for optimal Clifford circuit synthesis. In: Proceedings of the 28th Asia and South Pacific Design Automation Conference. pp. 190–195 (2023)
149. Seitz, P., Medina, I., Cruz, E., Huang, Q., Mendl, C.B.: Simulating quantum circuits using tree tensor networks. Quantum 7, 964 (2023)
150. Selinger, P.: A survey of graphical languages for monoidal categories. In: Coecke, B. (ed.) New Structures for Physics, pp. 289–355. Springer Berlin Heidelberg, Berlin, Heidelberg (2011), https://doi.org/10.1007/978-3-642-12821-9_4
151. Shaik, I., van de Pol, J.: Optimal layout synthesis for quantum circuits as classical planning. arXiv preprint arXiv:2304.12014 (2023)
152. Shannon, C.E.: The synthesis of two-terminal switching circuits. The Bell System Technical Journal 28(1), 59–98 (1949)
153. Shutty, N., Chamberland, C.: Decoding merged color-surface codes and finding fault-tolerant Clifford circuits using solvers for satisfiability modulo theories. Physical Review Applied 18(1), 014072 (2022)
154. Silva, J.M., Sakallah, K.A.: GRASP-a new search algorithm for satisfiability. In: Proceedings of International Conference on Computer Aided Design. pp. 220–227. IEEE (1996)

155. Sistla, M., Chaudhuri, S., Reps, T.: Symbolic quantum simulation with quasimodo. In: International Conference on Computer Aided Verification. pp. 213–225. Springer (2023)
156. Sistla, M., Chaudhuri, S., Reps, T.: Weighted context-free-language ordered binary decision diagrams. arXiv preprint arXiv:2305.13610 (2023)
157. Sistla, M.A., Chaudhuri, S., Reps, T.: CFLOBDDs: Context-free-language ordered binary decision diagrams. ACM Transactions on Programming Languages and Systems (2023)
158. Staudacher, K., Guggemos, T., Grundner-Culemann, S., Gehrke, W.: Reducing 2-qubit gate count for ZX-calculus based quantum circuit optimization. Electronic Proceedings in Theoretical Computer Science 394, 29–45 (Nov 2023), http://dx.doi.org/10.4204/EPTCS.394.3
159. Tafertshofer, P., Pedram, M.: Factored EVBDDs and their application to matrix representation and manipulation. Tech. rep., CENG Technical Report 94-27, Department of EE-Systems, University of Southern California (1994)
160. Tafertshofer, P., Pedram, M.: Factored edge-valued binary decision diagrams. Formal Methods in System Design 10(2), 243–270 (1997)
161. Tan, B., Cong, J.: Optimal layout synthesis for quantum computing. In: Proceedings of the 39th International Conference on Computer-Aided Design. pp. 1–9 (2020)
162. Tanaka, Y.: Exact non-identity check is NQP-complete. International Journal of Quantum Information 8(05), 807–819 (2010)
163. Tang, E.: A quantum-inspired classical algorithm for recommendation systems. In: Proceedings of the 51st annual ACM SIGACT symposium on theory of computing. pp. 217–228 (2019)
164. Terhal, B.M.: Quantum error correction for quantum memories. Rev. Mod. Phys. 87, 307–346 (Apr 2015), https://link.aps.org/doi/10.1103/RevModPhys.87.307
165. Thanos, D., Coopmans, T., Laarman, A.: Fast equivalence checking of quantum circuits of Clifford gates. In: André, É., Sun, J. (eds.) Automated Technology for Verification and Analysis. pp. 199–216. Springer Nature Switzerland, Cham (2023)
166. Thierry-Mieg, Y., Poitrenaud, D., Hamez, A., Kordon, F.: Hierarchical set decision diagrams and regular models. In: Tools and Algorithms for the Construction and Analysis of Systems: 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings 15. pp. 1–15. Springer (2009)
167. Toumi, A., Yeung, R., Felice, G.: Diagrammatic differentiation for quantum machine learning. Electronic Proceedings in Theoretical Computer Science 343, 132–144 (09 2021)
168. Venturelli, D., Do, M., O'Gorman, B., Frank, J., Rieffel, E., Booth, K.E.C., Nguyen, T., Narayan, P., Nanda, S.: Quantum circuit compilation: An emerging application for automated reasoning. In: Scheduling and Planning Applications Workshop (2019), https://openreview.net/forum?id=S1eEBO3nFE
169. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Improving gate-level simulation of quantum circuits. Quantum Information Processing 2(5), 347–380 (2003)
170. Viamontes, G.F., Markov, I.L., Hayes, J.P.: Quantum circuit simulation. Springer Science & Business Media (2009)

171. Viamontes, G., Markov, I., Hayes, J.: High-performance QuIDD-based simulation of quantum circuits. In: Proceedings Design, Automation and Test in Europe Conference and Exhibition. vol. 2, pp. 1354–1355 Vol.2 (2004)
172. Villalonga, B., Boixo, S., Nelson, B., Henze, C., Rieffel, E., Biswas, R., Mandrà, S.: A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. npj Quantum Information 5(1), 86 (2019)
173. Villoria, A., Basold, H., Laarman, A.: Enriching diagrams with algebraic operations. arXiv preprint arXiv:2310.11288 (2023)
174. Vilmart, R.: A near-optimal axiomatisation of ZX-calculus for pure qubit quantum mechanics. arXiv preprint arXiv:1812.09114 (2018), https://doi.org/10.48550/arXiv.1812.09114
175. Vilmart, R.: Quantum multiple-valued decision diagrams in graphical calculi. In: Bonchi, F., Puglisi, S.J. (eds.) 46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 202, pp. 89:1–89:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021), https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2021.89
176. Vinkhuijzen, L., Coopmans, T., Elkouss, D., Dunjko, V., Laarman, A.: LIMDD: A decision diagram for simulation of quantum computing including stabilizer states. Quantum 7, 1108 (2023), https://doi.org/10.22331/q-2023-09-11-1108
177. Vinkhuijzen, L., Coopmans, T., Laarman, A.: A knowledge compilation map for quantum information. arXiv preprint arXiv:2401.01322 (2024), https://doi.org/10.48550/arXiv.2401.01322
178. Vinkhuijzen, L., Grurl, T., Hillmich, S., Brand, S., Wille, R., Laarman, A.: Efficient implementation of LIMDDs for quantum circuit simulation. In: International Symposium on Model Checking of Software (SPIN) (2023)
179. Vrudhula, S.B.K., Pedram, M., Lai, Y.T.: Edge Valued Binary Decision Diagrams, pp. 109–132. Springer US (1996)
180. Wagner, F., Bärmann, A., Liers, F., Weissenbäck, M.: Improving quantum computation by optimized qubit routing. Journal of Optimization Theory and Applications 197(3), 1161–1194 (2023)
181. Wang, Q.: An algebraic axiomatisation of ZX-calculus. Electronic Proceedings in Theoretical Computer Science 340, 303–332 (Sep 2021), http://dx.doi.org/10.4204/EPTCS.340.16
182. Wang, Q., Yeung, R., Koch, M.: Differentiating and integrating ZX diagrams with applications to quantum machine learning (2022)
183. Wang, S.A., Lu, C.Y., Tsai, I.M., Kuo, S.Y.: An XQDD-based verification method for quantum circuits. IEICE transactions on fundamentals of electronics, communications and computer sciences 91(2), 584–594 (2008)
184. Wei, C.Y., Tsai, Y.H., Jhang, C.S., Jiang, J.H.R.: Accurate BDD-based unitary operator manipulation for scalable and robust quantum circuit verification. In: Proceedings of the 59th ACM/IEEE Design Automation Conference. pp. 523–528 (2022)
185. van de Wetering, J.: ZX-calculus for the working quantum computer scientist. arXiv preprint arXiv:2012.13966 (2020)
186. Wille, R., Burgholzer, L., Artner, M.: Visualizing decision diagrams for quantum computing (special session summary). In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 768–773. IEEE (2021)
187. Wille, R., Burgholzer, L., Hillmich, S., Grurl, T., Ploier, A., Peham, T.: The basis of design tools for quantum computing: arrays, decision diagrams, tensor net-

works, and ZX-calculus. In: Proceedings of the 59th ACM/IEEE Design Automation Conference. pp. 1367–1370. DAC '22, Association for Computing Machinery, New York, NY, USA (2022), https://doi.org/10.1145/3489517.3530627
188. Wille, R., Hillmich, S., Burgholzer, L.: Tools for quantum computing based on decision diagrams. ACM Transactions on Quantum Computing 3(3), 1–17 (2022)
189. Wille, R., Przigoda, N., Drechsler, R.: A compact and efficient SAT encoding for quantum circuits. In: 2013 Africon. pp. 1–6. IEEE (2013)
190. Wille, R., Zhang, H., Drechsler, R.: ATPG for reversible circuits using simulation, Boolean satisfiability, and pseudo Boolean optimization. In: 2011 IEEE Computer Society Annual Symposium on VLSI. pp. 120–125 (2011)
191. Wilson, N.: Decision diagrams for the computation of semiring valuations. In: Proceedings of the 19th international joint conference on Artificial intelligence. pp. 331–336 (2005)
192. Winderl, D., Huang, Q., Mendl, C.B.: A recursively partitioned approach to architecture-aware ZX polynomial synthesis and optimization. In: 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE (Sep 2023), http://dx.doi.org/10.1109/QCE57702.2023.00098
193. de Wolf, R.: Quantum computing: Lecture notes. arXiv preprint arXiv:1907.09415 (2019), https://doi.org/10.48550/arXiv.1907.09415
194. Wood, C.J., Biamonte, J.D., Cory, D.G.: Tensor networks and graphical calculus for open quantum systems. Quantum Info. Comput. 15(9–10), 759–811 (jul 2015)
195. Yamashita, S., Markov, I.L.: Fast equivalence-checking for quantum circuits. In: 2010 IEEE/ACM International Symposium on Nanoscale Architectures. pp. 23–28. IEEE (2010)
196. Ying, M.: Floyd–Hoare logic for quantum programs. ACM Transactions on Programming Languages and Systems (TOPLAS) 33(6), 1–49 (2012)
197. Zulehner, A., Hillmich, S., Wille, R.: How to efficiently handle complex values? implementing decision diagrams for quantum computing. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–7. IEEE (2019)
198. Zulehner, A., Wille, R.: Improving synthesis of reversible circuits: Exploiting redundancies in paths and nodes of QMDDs. In: Reversible Computation: 9th International Conference, RC 2017, Kolkata, India, July 6-7, 2017, Proceedings 9. pp. 232–247. Springer (2017)
199. Zulehner, A., Wille, R.: Advanced simulation of quantum computations. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38(5), 848–859 (2019)