Simulation and Verification of Quantum Circuits

Christian Schilling
Aalborg University
christianms@cs.aau.dk

AALBORG UNIVERSITY

DIGITAL TECH SUMMIT 2025

INDEPENDENT RESEARCH FUND DENMARK

DIREC Digital Research Centre Denmark

DeiC

Overview

Motivation of two fundamental problems

Simulation of quantum circuits

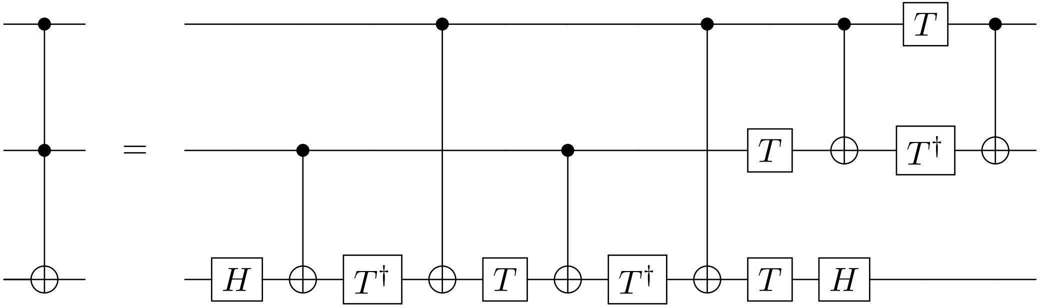
Formal verification for equivalence checking of quantum circuits

Conclusion

Overview

Circuit compilation (both conventional and quantum)

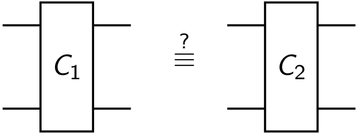
- During circuit design: use **high-level gates** and assume **arbitrary connectivity**
- **Compiler** translates to **low-level circuit** for executing on real hardware, supporting few **low-level gate types** and satisfying **connectivity constraints**



Circuit compilation (both conventional and quantum)

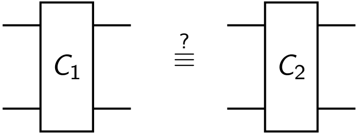
- During circuit design: use **high-level gates** and assume **arbitrary connectivity**
- **Compiler** translates to **low-level circuit** for executing on real hardware, supporting few **low-level gate types** and satisfying **connectivity constraints**
- In this process, compilers have lots of room for optimization:
  - Reduce amount of gates / operations
  - Quantum gates incur different levels of error (noise)
  - Deeper quantum circuits incur more errors
  - ...
- Important that **circuits are equivalent** before and after compilation (i.e., compute the same output for the same input)
  - How can we check **equivalence of circuits**?

Black-box equivalence check for **conventional** circuits



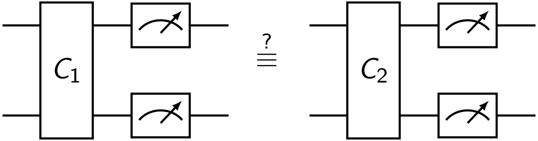
- Pick an input and execute both circuits
  - One-way result: Disagreement implies different circuits
  - **Exponentially many** classical inputs

Black-box equivalence check for **quantum** circuits



- Pick an input and execute both circuits
  - One-way result: Disagreement implies different circuits
  - **Exponentially many** classical inputs
- Infinitely many quantum states as input?
  - **Sufficient to check basis states**
  - **Exponentially many** basis states

Black-box equivalence check for **quantum** circuits



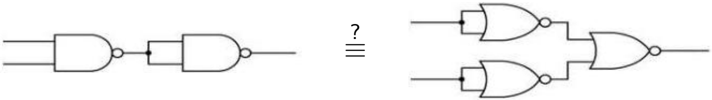
- Pick an input and execute both circuits
  - One-way result: Disagreement implies different circuits
  - **Exponentially many** classical inputs
- Infinitely many quantum states as input?
  - **Sufficient to check basis states**
  - **Exponentially many** basis states
- **Can only observe basis states (via measurement)**
  - Disagreement does not imply different circuits
  - Statistical result by executing many times – **even more expensive**

Equivalence checking of circuits

- Two general directions: **testing/sampling** and **formal verification**
- **Testing**: choose an input and run the circuit
  - Single test runs are cheap, but result is **not conclusive**
  - Quantum circuits:
    - Expensive and hardly available (yet)
    - Result is probabilistic $\rightsquigarrow$ many runs for same input needed
    - Instead can **simulate circuit on conventional computer**
- **Formal verification**: mathematical proof that circuits are equivalent
  - More expensive than a few test runs, but result is **conclusive**
  - Run **on conventional computer**

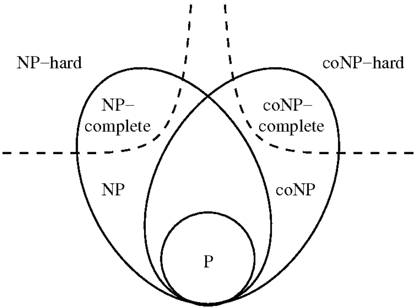
White-box equivalence check for **conventional** circuits

- What if we know the conventional circuits?



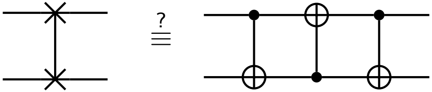
White-box equivalence check for **conventional** circuits

- Checking that two conventional circuits are equivalent is co-NP-complete
  - Believed to require exponential complexity
  - So in principle not better than checking all possible inputs



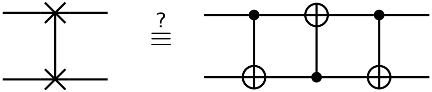
White-box equivalence check for **quantum** circuits

- What if we know the quantum circuits?



White-box equivalence check for **quantum** circuits

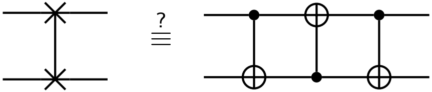
- What if we know the quantum circuits?



- Can simulate all basis states $\rightsquigarrow$ algorithm with definite result

White-box equivalence check for **quantum** circuits

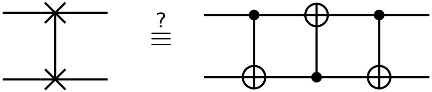
- What if we know the quantum circuits?



- Can simulate all basis states $\rightsquigarrow$ algorithm with definite result
  - **Exponentially** many basis states
  - Each simulation takes **exponential time**

White-box equivalence check for **quantum** circuits

- What if we know the quantum circuits?

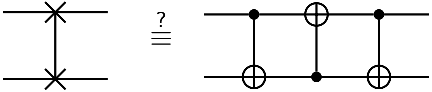


- Can simulate all basis states $\rightsquigarrow$ algorithm with definite result
  - **Exponentially** many basis states
  - Each simulation takes **exponential time**
- Alternative: compare characteristic matrices
  - Matrices are **exponentially large**

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \stackrel{?}{=} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

White-box equivalence check for **quantum** circuits

- What if we know the quantum circuits?



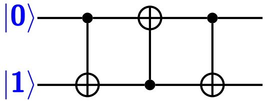
- Can simulate all basis states $\rightsquigarrow$ algorithm with definite result
  - **Exponentially** many basis states
  - Each simulation takes **exponential time**
- Alternative: compare characteristic matrices
  - Matrices are **exponentially large**
- No way around: problem is co-NQP-complete[1]
  - Believed to require exponential complexity

[1] Y. Tanaka. *Int. J. Quantum Inf.* (2010)

Overview

Simulation on conventional computer

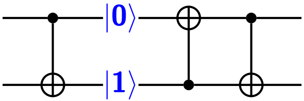
- Simplest approach: propagate (exponentially large) state vector



$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}$$

Simulation on conventional computer

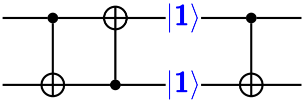
- Simplest approach: propagate (exponentially large) state vector



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

Simulation on conventional computer

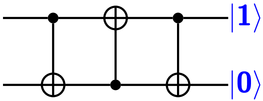
- Simplest approach: propagate (exponentially large) state vector



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

Simulation on conventional computer

- Simplest approach: propagate (exponentially large) state vector

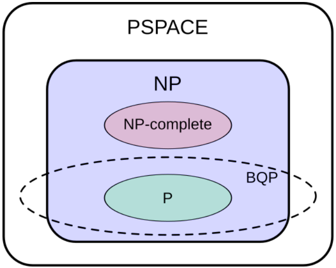


$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- Now we actually have the **quantum state** (no measurements required)

Complexity of simulating a quantum circuit

- Simulation is BQP-complete
  - Believed to require exponential complexity (on conventional computer)
- **Clifford gates** (Hadamard, CNOT, phase S) can be simulated **efficiently**[1]
  - Non-universal gate set
  - Relevant for **error correction**, which will play central role in fault-tolerant era
  - **Equivalence checking** is also efficient[2]



[1] D. Gottesman. PhD thesis. 1997

[2] D. Thanos, T. Coopmans, and A. Laarman. *ATVA*. 2023

Simulation based on tensor networks

- Each step of the matrix-vector multiplications had exponential complexity
- Idea behind **tensor networks**: perform cheaper calculations when possible

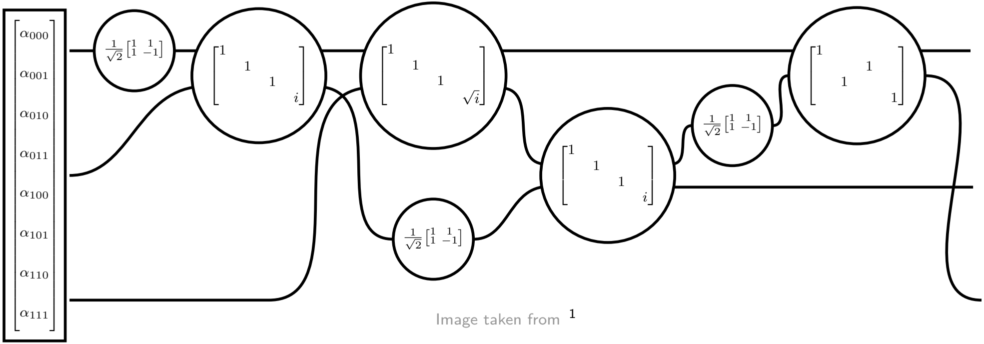


Image taken from [1]

[1] (L. Burgholzer, A. Ploier, and R. Wille. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* [2023])

Simulation based on tensor networks

- **Tensor network**: graph of tensors, initially corresponding to gates in circuit
- Nodes with shared edges can be **contracted** (= merged) **in any order**

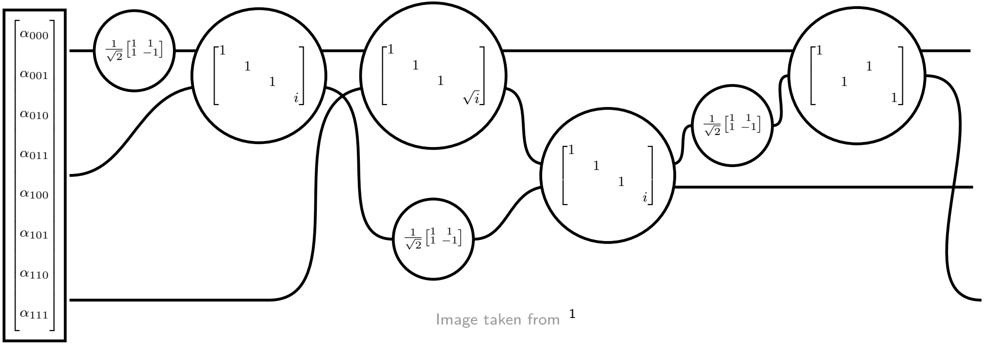


Image taken from [1]

[1] (L. Burgholzer, A. Ploier, and R. Wille. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* [2023])

Simulation based on tensor networks

- **Tensor network**: graph of tensors, initially corresponding to gates in circuit
- Nodes with shared edges can be **contracted** (= merged) **in any order**
- Finding the optimal contraction order is NP-hard[1]
- Practice: use of good enough and efficient solutions (heuristics)[2]

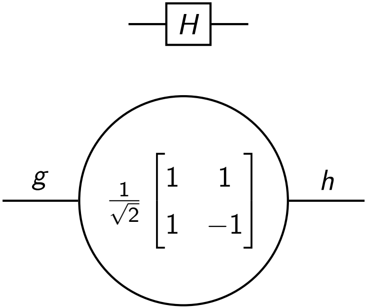
---

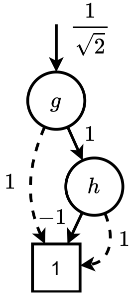
[1] C. Lam, P. Sadayappan, and R. Wenger. *Parallel Process. Lett.* (1997).
[2] J. Gray and S. Kourtis. *Quantum* (2021).

Tensor decision diagrams (TDDs)[1]

- Alternative representation of a tensor
- Sometimes avoids exponential size of matrix / vector representation

Example: Hadamard gate with tensor and tensor decision diagram





[1]X. Hong, X. Zhou, S. Li, Y. Feng, and M. Ying. *ACM Trans. Design Autom. Electr. Syst.* (2022).

Overview

Motivation of two fundamental problems

Simulation of quantum circuits

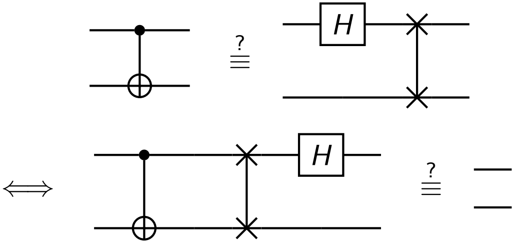
Formal verification for equivalence checking of quantum circuits

Conclusion

Reverse scheme for equivalence checking[1]

$$C_1 \equiv C_2 \stackrel{\text{def}}{\Longleftrightarrow} \exists\theta\colon U_1 = e^{i\theta}\cdot U_2 \Longleftrightarrow \exists\theta\colon U_1 \cdot U_2^\dagger = e^{i\theta}\cdot I \stackrel{\text{def}}{\Longleftrightarrow} C_1 C_2^{-1} \equiv C_I$$

- C_2^{-1} is the inverted C_2 (reversed and each gate inverted)



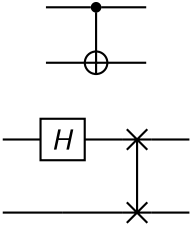
(Coincidentally, the swap and Hadamard gates are self-inverse)

[1] G. F. Viamontes, I. L. Markov, and J. P. Hayes. *ICCAD*. 2007.

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2



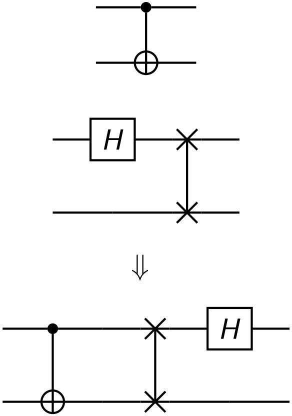
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$



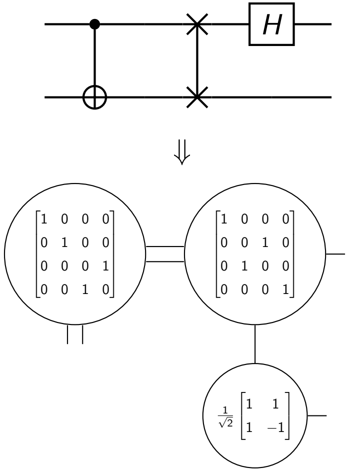
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$
2. Convert $C_1 C_2^{-1}$ to tensor network



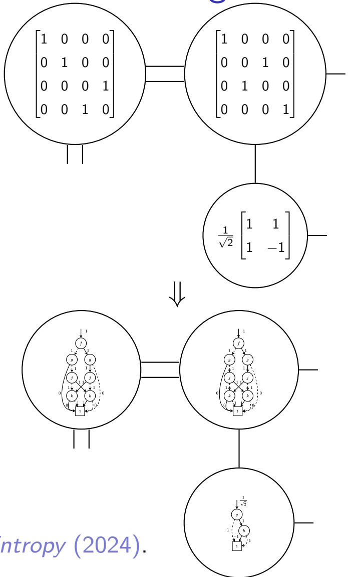
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$
2. Convert $C_1 C_2^{-1}$ to tensor network
3. Convert all tensors to TDDs



(TDDs on the right are only exemplary)

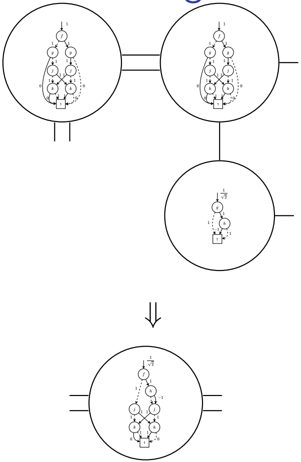
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$
2. Convert $C_1 C_2^{-1}$ to tensor network
3. Convert all tensors to TDDs
4. Contract TDD network

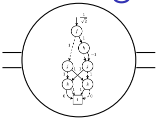


(TDDs on the right are only exemplary)

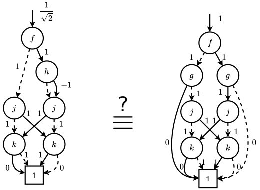
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs



$\Downarrow$



Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$
2. Convert $C_1 C_2^{-1}$ to tensor network
3. Convert all tensors to TDDs
4. Contract TDD network
5. Compare final TDD to identity TDD

(TDDs on the right are only exemplary)

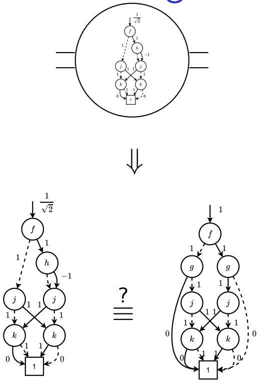
[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

TDD-based algorithm for equivalence checking[1]

Algorithm combines reverse scheme, tensor networks, and TDDs

Given: Quantum circuits C_1, C_2

1. Construct circuit $C_1 C_2^{-1}$
2. Convert $C_1 C_2^{-1}$ to tensor network
3. Convert all tensors to TDDs
4. **Contract TDD network**
5. Compare final TDD to identity TDD



(TDDs on the right are only exemplary)

[1] C. B. Larsen, S. B. Olsen, K. G. Larsen, and C. Schilling. *Entropy* (2024).

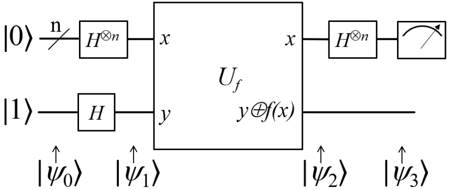
Empirical evaluation on circuits from three quantum algorithms

- Circuits from MQT Bench[1] with varying number of qubits at two compilation levels (level 1 and 3 (out of 4)) with significantly different gate sets and layouts
  - Deutsch-Jozsa algorithm (DJ)
  - Greenberger-Horne-Zeilinger state preparation (GHZ)
  - Graph state preparation (GS)

[1] N. Quetschlich, L. Burgholzer, and R. Wille. *Quantum* (2023).

Deutsch-Jozsa algorithm (DJ)[1,2]

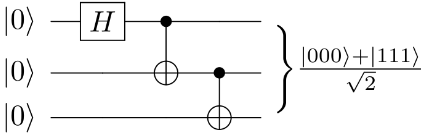
- Given $f: \{0,1\}^n \to \{0,1\}$ with promise that it is either
  - constant (100% "0" or 100% "1") or
  - balanced (50% "0" and 50% "1")
- Task: Determine which of the two cases it is
- Demonstrates exponential speed-up (requires a single shot)



[1] D. Deutsch and R. Jozsa. *Proc. R. Soc. A* (1992).
[2] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. *Proc. R. Soc. A* (1998).

Greenberger-Horne-Zeilinger state preparation (GHZ)[1]

- The **GHZ state** generalizes the Bell state
- For 3 qubits: $\dfrac{|000\rangle + |111\rangle}{\sqrt{2}}$
- For k qubits: $\dfrac{|0\rangle^{\otimes k} + |1\rangle^{\otimes k}}{\sqrt{2}}$
- Used in quantum communication and cryptography protocols



[1] D. M. Greenberger, M. A. Horne, and A. Zeilinger. *Bell's theorem, quantum theory and conceptions of the universe.* 1989.

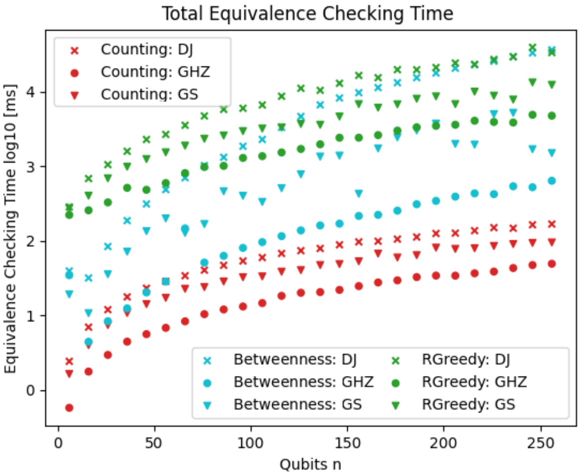
Graph state preparation (GS)[1]

- A **graph state** is a state that can be represented by a graph
- Each vertex corresponds to a qubit
- $|G\rangle = \prod_{(u,v)\in E} CZ^{(u,v)} |+\rangle^{\otimes |V|}$

  where $CZ^{(u,v)}$ is the corresponding controlled-Z gate
- Useful, e.g., in quantum error-correcting codes

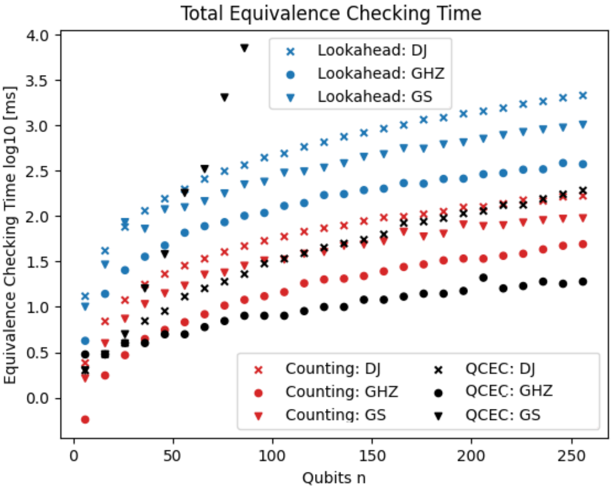
[1] M. Hein et al. *arXiv preprint quant-ph/0602096* (2006).

Comparison to cotengra[1]



[1] J. Gray and S. Kourtis. *Quantum* (2021).

Comparison to QCEC[1]



[1] L. Burgholzer and R. Wille. *Softw. Impacts* (2021).

Overview

Motivation of two fundamental problems

Simulation of quantum circuits

Formal verification for equivalence checking of quantum circuits

Conclusion

Conclusion

- **Equivalence checking** is a central problem
  - Both for conventional and quantum computers
  - Theoretically intractable, but practical solutions often work
- **Simulation** and **formal verification** are powerful technologies
- **Promising tools** are being developed
  - Tensor networks
  - Decision diagrams
  - . . . many more!

More about quantum from Aalborg University

- At Digital Tech Summit
  - Aalborg University booth @ UNI3
  - Petar Popovski: *Low-Latency Classical Communications for Quantum Applications* (tomorrow 9:30)
- In 2026
  - Hosting Danisch Quantum Community's *Scientific Quantum Conference*
  - Hosting IEEE *Int. Conference on Quantum Control, Computing and Learning*
  - Organizing Workshop on *Formal Methods in Quantum Computing*
- In general
  - AAU Quantum Hub
  - CLASSIQUE: *Center for Classical Communication in the Quantum Era*