Bounds for Quantum Circuits using Logic-Based Analysis

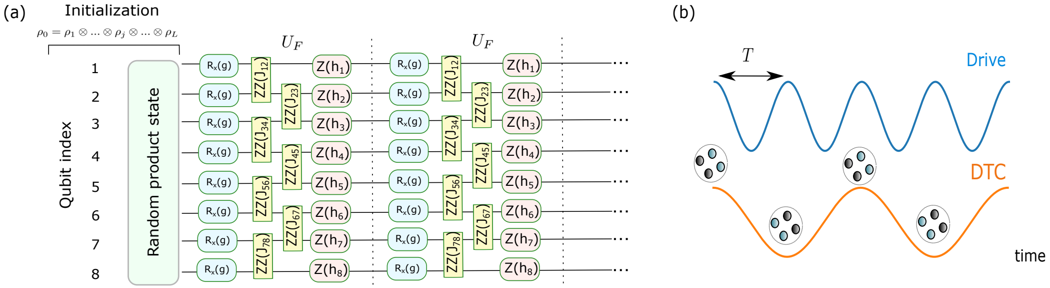
Benedikt Fauseweh, Ben Hermann, and **Falk Howar**

TU Dortmund University

Feb 24, 2025

technische universität
dortmund

Many-Body Localized Discrete Time Crystals



- Non-equilibrium phase of matter characterized by a spontaneous breaking of discrete time-translation symmetry, resulting in a subharmonic response that spontaneously breaks the periodicity of an external drive
- Despite significant interest, the existence of MBL-DTCs remains an open question due to the potential instability of the underlying MBL phase

A more complete discussion in the paper ...

Analysis of Quantum Circuits

QSE Challenge

Circuit developers want to design circuits that stay within correct sub-space

- Reasoning non-trivial, requires deep insight into mechanics of quantum program and underlying theory
- Showing bounds would reduce need for full quantum simulation
- But: No methods to proof that a circuit stays within sub-space, yet

Analysis of Quantum Circuits

QSE Challenge

Circuit developers want to design circuits that stay within correct sub-space

- Reasoning non-trivial, requires deep insight into mechanics of quantum program and underlying theory
- Showing bounds would reduce need for full quantum simulation
- But: No methods to proof that a circuit stays within sub-space, yet

Idea for a Solution

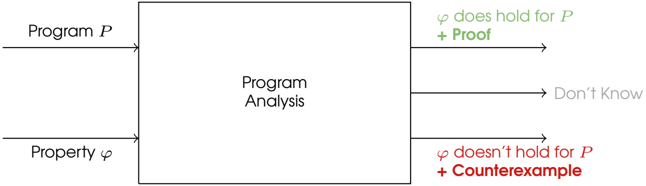
Adapt and **scale** symbolic verification techniques to quantum circuits

- Today's quantum software formulated as circuits
- Automated Reasoning and symbolic techniques had big impact in (classic) hardware verification
- After hardware: big impact on software (e.g. driver verification at MicroSoft)

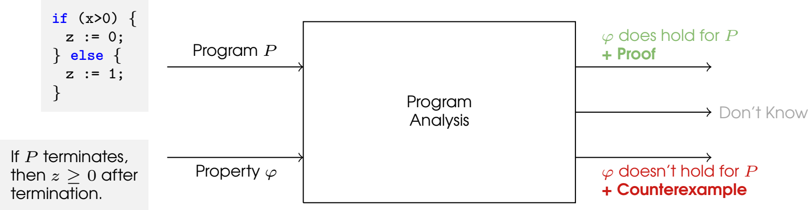
Outline

- Logic-based analysis of quantum circuits and challenges
- Tactics for scaling verification
- Initial results

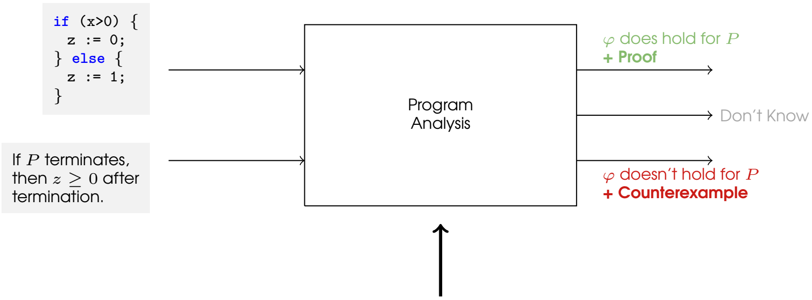
Program Analysis



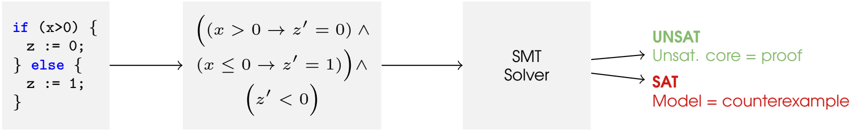
Program Analysis



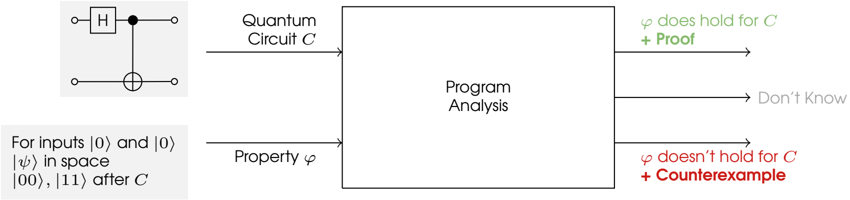
Program Analysis



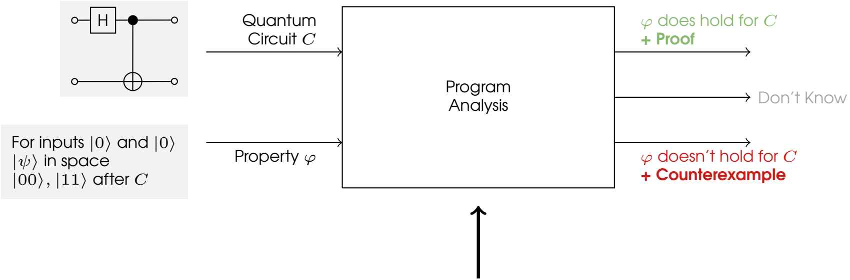
Many different techniques exist. Here: Encoding as SMT problem $P \land \neg\varphi$:



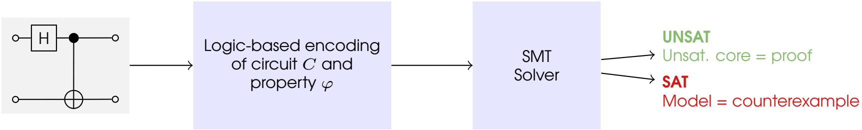
Program Analysis



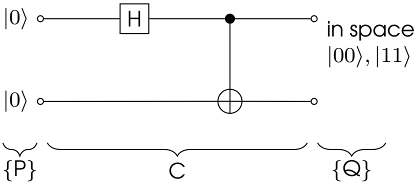
Program Analysis



Different techniques exist. Here: Encoding as SMT problem $C \wedge \neg\varphi$:



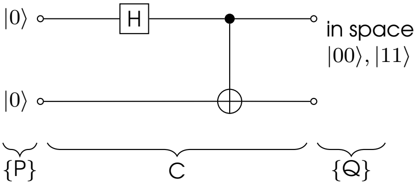
Logic-Based Encoding of Quantum Circuits



$$|\psi\rangle \ := \ \mathbf{c_{00}}|00\rangle + \mathbf{c_{01}}|01\rangle + \mathbf{c_{10}}|10\rangle + \mathbf{c_{11}}|11\rangle$$

Cf. Bauer-Marquart et al., FM 2023

Logic-Based Encoding of Quantum Circuits



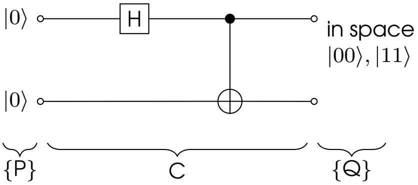
$$|\psi\rangle := \mathbf{c_{00}}|00\rangle + \mathbf{c_{01}}|01\rangle + \mathbf{c_{10}}|10\rangle + \mathbf{c_{11}}|11\rangle$$

$$H := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$CNOT := \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

Cf. Bauer-Marquart et al., FM 2023

Logic-Based Encoding of Quantum Circuits



$$|\psi\rangle \;:=\; \mathbf{c_{00}}|00\rangle + \mathbf{c_{01}}|01\rangle + \mathbf{c_{10}}|10\rangle + \mathbf{c_{11}}|11\rangle$$

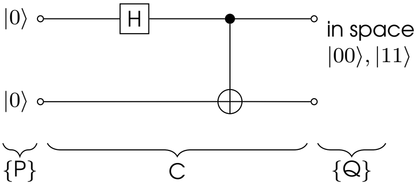
$$P := (\mathbf{c^0_{00}} = 1) \;\wedge\; (\mathbf{c^0_{01}} = 0) \;\wedge\; (\mathbf{c^0_{10}} = 0) \;\wedge\; (\mathbf{c^0_{11}} = 0)$$

$$H \;:=\; \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$CNOT \;:=\; \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Cf. Bauer-Marquart et al., FM 2023

Logic-Based Encoding of Quantum Circuits



$$|\psi\rangle \;:=\; \mathbf{c_{00}}|00\rangle + \mathbf{c_{01}}|01\rangle + \mathbf{c_{10}}|10\rangle + \mathbf{c_{11}}|11\rangle$$

$$P := (\mathbf{c^0_{00}} = 1) \,\wedge\, (\mathbf{c^0_{01}} = 0) \,\wedge\, (\mathbf{c^0_{10}} = 0) \,\wedge\, (\mathbf{c^0_{11}} = 0)$$

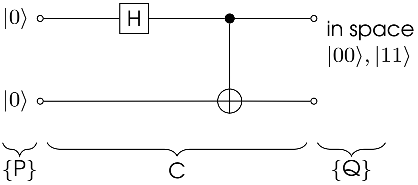
$$\begin{aligned} C := {} & (\mathbf{c^2_{00}} = \mathbf{c^1_{00}}) \wedge (\mathbf{c^1_{00}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{00}} + \mathbf{c^0_{10}})) \wedge \\ & (\mathbf{c^2_{01}} = \mathbf{c^1_{01}}) \wedge (\mathbf{c^1_{01}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{01}} + \mathbf{c^0_{11}})) \wedge \\ & (\mathbf{c^2_{10}} = \mathbf{c^1_{11}}) \wedge (\mathbf{c^1_{10}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{00}} - \mathbf{c^0_{10}})) \wedge \\ & (\mathbf{c^2_{11}} = \mathbf{c^1_{10}}) \wedge (\mathbf{c^1_{11}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{01}} - \mathbf{c^0_{11}})) \end{aligned}$$

$$H \;:=\; \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$CNOT \;:=\; \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

Cf. Bauer-Marquart et al., FM 2023

Logic-Based Encoding of Quantum Circuits



$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$CNOT := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$|\psi\rangle := \mathbf{c_{00}}|00\rangle + \mathbf{c_{01}}|01\rangle + \mathbf{c_{10}}|10\rangle + \mathbf{c_{11}}|11\rangle$$

$$P := (\mathbf{c^0_{00}} = 1) \wedge (\mathbf{c^0_{01}} = 0) \wedge (\mathbf{c^0_{10}} = 0) \wedge (\mathbf{c^0_{11}} = 0)$$

$$\begin{aligned} C := (\mathbf{c^2_{00}} = \mathbf{c^1_{00}}) \wedge (\mathbf{c^1_{00}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{00}} + \mathbf{c^0_{10}})) \wedge \\ (\mathbf{c^2_{01}} = \mathbf{c^1_{01}}) \wedge (\mathbf{c^1_{01}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{01}} + \mathbf{c^0_{11}})) \wedge \\ (\mathbf{c^2_{10}} = \mathbf{c^1_{11}}) \wedge (\mathbf{c^1_{10}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{00}} - \mathbf{c^0_{10}})) \wedge \\ (\mathbf{c^2_{11}} = \mathbf{c^1_{10}}) \wedge (\mathbf{c^1_{11}} = \tfrac{1}{\sqrt{2}}(\mathbf{c^0_{01}} - \mathbf{c^0_{11}})) \end{aligned}$$

$$Q := (\mathbf{c^2_{01}} = 0) \wedge (\mathbf{c^2_{10}} = 0)$$

Cf. Bauer-Marquart et al., FM 2023

Challenges

- **Size of logic encoding exponential in number of qubits**
  - k-qubit state described by 2^k **complex** coefficients
  - 2^k **complex** coefficients can be modeled by 2^{k+1} **real coefficients and nonlinear real arithmetic**

Different from challenges in classic program verification: loops, function calls, memory allocation, concurrency

Challenges

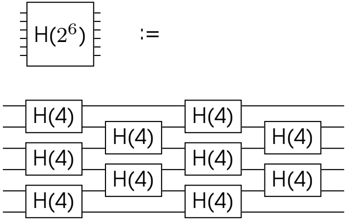
- **Size of logic encoding exponential in number of qubits**
  - k-qubit state described by 2^k **complex** coefficients
  - 2^k **complex** coefficients can be modeled by 2^{k+1} **real coefficients and nonlinear real arithmetic**
- **Some gate effects are described by elementary functions**
  - Rotations are described by **trigonometric functions**
  - Hadamard is described using $\sqrt{\mathbf{2}}$

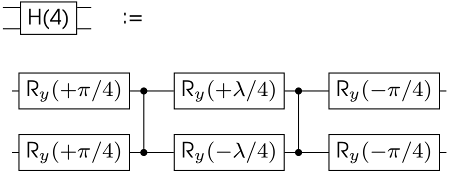
Different from challenges in classic program verification: loops, function calls, memory allocation, concurrency

Two tactics for scaling verification:

decomposition and **abstraction / over-approximation**

A Slightly Bigger Example: $H(2^6)$





Structure:

- 6 qubits = 64 complex coefficients = 128 real state variables
- Hierarchical composition of 10 $H(4)$ circuits
- Each $H(4)$: 6 rotations and 2 CZ gates
- Rotations parameterized by λ

Properties:

- $H(2^6)$ preserves expected Hamming weight
- $H(4)$ preserves expected Hamming weight

$$\mathrm{HW}[\psi_{in}] = \mathrm{HW}[\psi_{out}]$$

$$\mathrm{HW}(\ |\psi\rangle\ ) = \sum_{i=0}^{2^n-1} w(i) \cdot |c_i|^2$$

Example from Anselmetti et al., New Journal of Physics, 2021

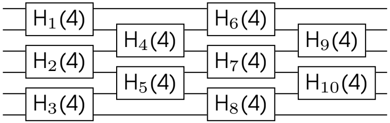
Tactic 1: Decomposition

Compositional Verification

- C sequential composition of sub-circuits $C_1, \ldots, C_n$.
- Local properties $A_1, \ldots, A_n$ such that
  - $C_i \models A_i$ for $1 \leq i < n$, and
  - $A_1 \wedge \ldots \wedge A_n \models \varphi$.
- Schema establishes $C_1 \wedge \ldots \wedge C_n \models \varphi$

(In the paper we show a compositional verification scheme for pre- and post-conditions)

Tactic 1: Decomposition



Compositional Verification

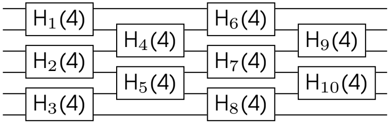
- C sequential composition of sub-circuits $C_1, \ldots, C_n$.
- Local properties $A_1, \ldots, A_n$ such that
  - $C_i \models A_i$ for $1 \leq i < n$, and
  - $A_1 \wedge \ldots \wedge A_n \models \varphi$.
- Schema establishes $C_1 \wedge \ldots \wedge C_n \models \varphi$

(In the paper we show a compositional verification scheme for pre- and post-conditions)

Tactic 1: Decomposition

Compositional Verification

- C sequential composition of sub-circuits $C_1, \ldots, C_n$.
- Local properties $A_1, \ldots, A_n$ such that
  - $C_i \models A_i$ for $1 \leq i < n$, and
  - $A_1 \wedge \ldots \wedge A_n \models \varphi$.
- Schema establishes $C_1 \wedge \ldots \wedge C_n \models \varphi$

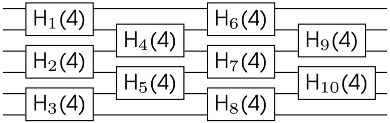


For $1 \leq i \leq 10$:

- $C_i := H_i(4)$
- $A_i := \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i]$

(In the paper we show a compositional verification scheme for pre- and post-conditions)

Tactic 1: Decomposition



Compositional Verification

- C sequential composition of sub-circuits $C_1, \ldots, C_n$.
- Local properties $A_1, \ldots, A_n$ such that
  - $C_i \models A_i$ for $1 \leq i < n$, and
  - $A_1 \wedge \ldots \wedge A_n \models \varphi$.
- Schema establishes $C_1 \wedge \ldots \wedge C_n \models \varphi$

For $1 \leq i \leq 10$:

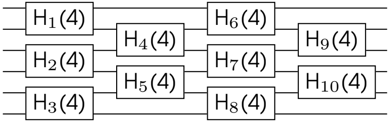
- $C_i := H_i(4)$
- $A_i := \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i]$
- $H_i(4) \models \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i]$
- $\bigwedge_i \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i] \models \mathrm{HW}[\psi_0] = \mathrm{HW}[\psi_{10}]$

(In the paper we show a compositional verification scheme for pre- and post-conditions)

Tactic 1: Decomposition

Compositional Verification

- C sequential composition of sub-circuits $C_1, \ldots, C_n$.
- Local properties $A_1, \ldots, A_n$ such that
  - $C_i \models A_i$ for $1 \leq i < n$, and
  - $A_1 \wedge \ldots \wedge A_n \models \varphi$.
- Schema establishes $C_1 \wedge \ldots \wedge C_n \models \varphi$



For $1 \leq i \leq 10$:

- $C_i := H_i(4)$
- $A_i := \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i]$

- $H_i(4) \models \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i]$
- $\bigwedge_i \mathrm{HW}[\psi_{i-1}] = \mathrm{HW}[\psi_i] \models \mathrm{HW}[\psi_0] = \mathrm{HW}[\psi_{10}]$

Establishes $H_i(2^6) \models \mathrm{HW}[\psi_0] = \mathrm{HW}[\psi_{10}]$

(In the paper we show a compositional verification scheme for pre- and post-conditions)

Leveraging Additional Lemmata

In the n-qubit system, the total expected Hamming weight can be written as:

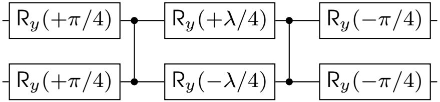
$$\mathrm{HW}[\psi] \ := \sum_{k \neq i,j} \langle\psi| \frac{1 - Z_k}{2} |\psi\rangle + \langle\psi| \frac{1 - Z_i}{2} + \frac{1 - Z_j}{2} |\psi\rangle$$

As a result, it suffices to proof $H_i(4) \ \models \ \mathrm{HW}[\psi_{in}] = \mathrm{HW}[\psi_{out}]$ on a 2-qubit state

Tactic 2: Abstraction and Over-Approximation

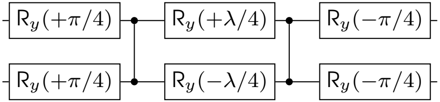
- **(Precise) Abstraction**
  - Summarize effect of multiple gates in simplified form
  - Proof obligation: simplified form equivalent to concrete representation
- **Over-Approximation**
  - Replace complex representation by over-approximation
  - May produce spurious counterexamples

Tactic 2: Abstraction and Over-Approximation



- **(Precise) Abstraction**
  - Summarize effect of multiple gates in simplified form
  - Proof obligation: simplified form equivalent to concrete representation
- **Over-Approximation**
  - Replace complex representation by over-approximation
  - May produce spurious counterexamples

Tactic 2: Abstraction and Over-Approximation



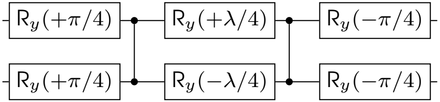
- **(Precise) Abstraction**
  - Summarize effect of multiple gates in simplified form
  - Proof obligation: simplified form equivalent to concrete representation
- **Over-Approximation**
  - Replace complex representation by over-approximation
  - May produce spurious counterexamples

Abstraction:

$$H(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & +s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{for} \quad \begin{aligned} c &:= \cos(\lambda/2) \\ s &:= \sin(\lambda/2) \end{aligned}$$

Tactic 2: Abstraction and Over-Approximation

- **(Precise) Abstraction**
  - Summarize effect of multiple gates in simplified form
  - Proof obligation: simplified form equivalent to concrete representation
- **Over-Approximation**
  - Replace complex representation by over-approximation
  - May produce spurious counterexamples



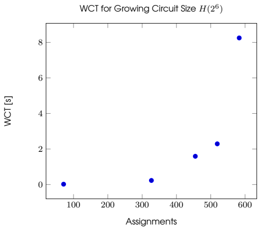
Abstraction:

$$H(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & +s & 0 \\ 0 & -s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{for} \quad \begin{array}{rcl} c & := & \cos(\lambda/2) \\ s & := & \sin(\lambda/2) \end{array}$$

Over-Approximation:

We only require that $0 \leq s, c \leq 1$ and that $s^2 + c^2 = 1$

Initial Results



Results for examples in slides:

- Techniques work
- Do not increase efficiency on H+CNOT
- Enable verification for $H(2^6)$
- Projection to 2-qubit state increases efficiency by order of magnitude

Example	Encoding			Analysis	
	Vars	Ass.	Logic	Res.	wct [s]
H+CNOT	25	26	LRA†	✓	0.005
H+CNOT, C1	17	11	LRA†	✓	0.005
H+CNOT, C2	17	11	LRA†	✓	0.003
H+CNOT, P+A1	9	3	LRA†	✓	0.004
H(2^6), naive	10 370	5 191	TRIG	-	DNS
H(2^6), precise	3 330	1 671	TRIG	-	DNS
H(2^6)	1 412	647	NRA	d/k	DNF
H(2^6), 9/10	1 284	583	NRA	✓	8.25
H(2^6), 8/10	1 156	519	NRA	✓	2.29
H(2^6), 7/10	1 028	. 455	NRA	✓	1.59
H(2^6), 5/10	772	. 327	NRA	✓	0.23
H(2^6), 1/10	260	71	NRA	✓	0.02
H(4)	20	15	NRA	✓	0.01

†: over-approximated $1/\sqrt{2}$, **DNS**: did not attempt to solve, **DNF**: timeout after 30 min

Summary, Open Questions, and Future Work

Summary:

- Logic-based verification for quantum circuits with hierarchical structure
- Scalability through compositional verification, abstraction, and over-approximation

Initial Results:

- Techniques applicable to studied circuits
- Techniques increase performance significantly

Summary, Open Questions, and Future Work

Summary:

- Logic-based verification for quantum circuits with hierarchical structure
- Scalability through compositional verification, abstraction, and over-approximation

Initial Results:

- Techniques applicable to studied circuits
- Techniques increase performance significantly

Open Questions (Decomposition):

- Can we automate generation of assumptions?
- Can we generate useful decompositions from
  - hierarchical circuit design,
  - static analysis (e.g. clone detection),
  - data flow analysis?

Open Questions (Scalability):

- Can the approach be automated or will it have to be interactive?
- Potential of abstraction and over-approximation?
- More lemmata that enable projection to sub-circuits?

Compositional Verification for H-CNOT Example

$$P := (c^0_{00} = 1) \wedge (c^0_{01} = 0) \wedge (c^0_{10} = 0) \wedge (c^0_{11} = 0)$$

Compositional Argument:

Schema for pre- and post-conditions:

$$A_1 := (\frac{1}{\sqrt{2}}(c^0_{01} + c^0_{11}) = 0) \wedge (\frac{1}{\sqrt{2}}(c^0_{01} - c^0_{11}) = 0)$$

$$\begin{aligned} P &\models A_1 \\ A_1 \wedge H &\models A_2 \\ A_2 \wedge CNOT &\models Q \\ \mathbf{P} \wedge \mathbf{H} \wedge \mathbf{CNOT} &\models \mathbf{Q} \end{aligned}$$

$$\begin{aligned} H := (c^1_{00} = \frac{1}{\sqrt{2}}(c^0_{00} + c^0_{10})) \wedge (c^1_{01} = \frac{1}{\sqrt{2}}(c^0_{01} + c^0_{11})) \wedge \\ (c^1_{10} = \frac{1}{\sqrt{2}}(c^0_{00} - c^0_{10})) \wedge (c^1_{11} = \frac{1}{\sqrt{2}}(c^0_{01} - c^0_{11})) \end{aligned}$$

Over-Approximation:

Use c and assumption $c \neq 0$ instead of $\sqrt{2}$

$$A_2 := (c^1_{01} = 0) \wedge (c^1_{11} = 0)$$

$$CNOT := (c^2_{00} = c^1_{00}) \wedge (c^2_{01} = c^1_{01}) \wedge (c^2_{10} = c^1_{11}) \wedge (c^2_{11} = c^1_{10})$$

$$Q := (c^2_{01} = 0) \wedge (c^2_{10} = 0)$$