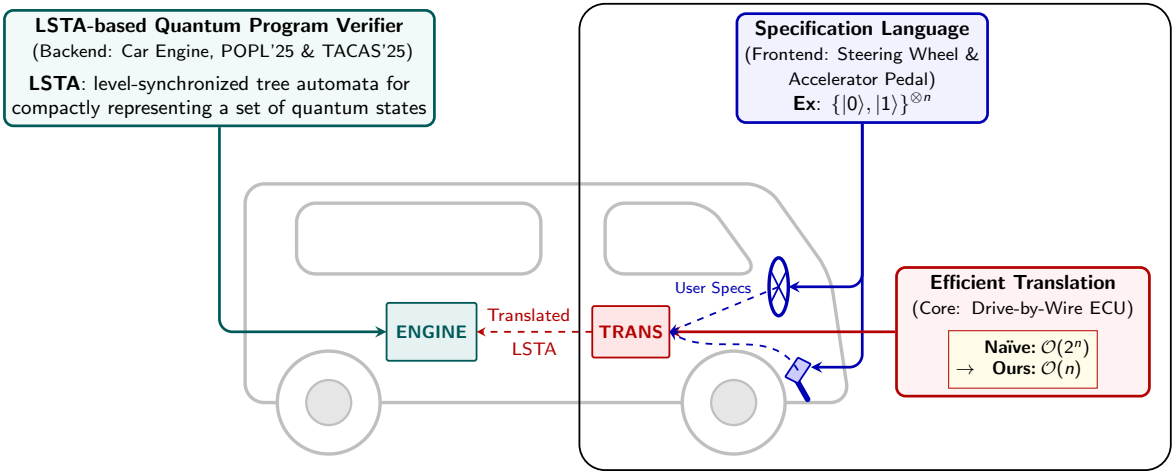
A Practical Specification Language for Automatic Quantum Program Verification

Wei-Lun Tsai[1,2] Yu-Fang Chen[1] Ondřej Lengál[3]

[1]Institute of Information Science, Academia Sinica, Taiwan
[2]Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan
[3]Faculty of Information Technology, Brno University of Technology, Czech Republic

CAV 2026

Overview - What We Have Done in Automatic Quantum Verification



The AutoQ Verification Framework: Verifying Parallel Hadamards

- **Hoare Triples**: $\{\mathbf{P}\}\ C\ \{\mathbf{Q}\}$
- Assertions (P and Q) are **sets of quantum states**.

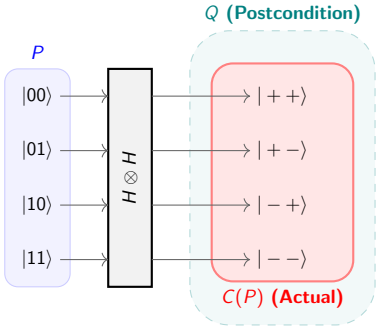
Step 1: Specification

- **Precondition** P: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$
- **Circuit** C: Parallel Hadamards ($H \otimes H$)
- **Postcondition** Q: Desired output set.

Step 3: Verification Condition

- **Success** if $C(P) \subseteq Q$.

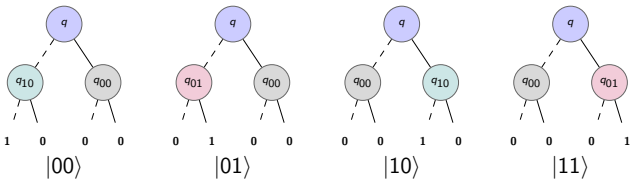
Step 2. State Evolution Mapping



How LSTA Encodes Sets of Quantum States

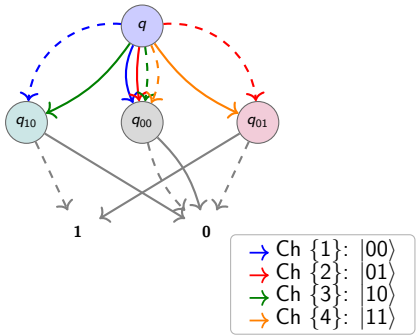
- The compactness of P and Q decides the verification efficiency.
- To encode a **set** of states, an LSTA uses **choices** to select different trees, while **merging common subtrees** to save space, which is the key to memory compression.

4 Distinct Trees (Naïve Approach)



Result: $\mathbf{3 \times 4 = 12}$ internal nodes.
Redundant subtrees (q_{00}, q_{01}, and q_{10}) are repeatedly created.

Single LSTA for Set P



Notice how these subtrees are heavily shared across all 4 choices!

The Bottleneck: Writing Automata is Hard

- The Usability Bottleneck: Although LSTAs provide incredible compression, manual construction of tree automata is way too far from how quantum researchers actually think.

Our Solution: A High-Level Specification Language

We design a grammar **tailored** to **mirror the standard mathematical representation** of sets of quantum states:

1. **Modular Construction:** Matches structural composition ($\otimes, \cup$) used in papers.
2. **Symbolic Representation:** Mirrors the notational brevity of Dirac $|\cdot\rangle$ and $\sum$.
3. **Constraint-Based Spec:** Precisely bounds the state space via standard complex arithmetic.

- We want researchers to write what they already write in their papers.

Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\{\ \bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\{\ \bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\{\ \bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0^n0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} \{\ a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} \{\ a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} \{\ a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} \{\ a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

Important Observation: The specifications are now **direct mathematical descriptions** of quantum properties, which are highly intuitive for quantum researchers.

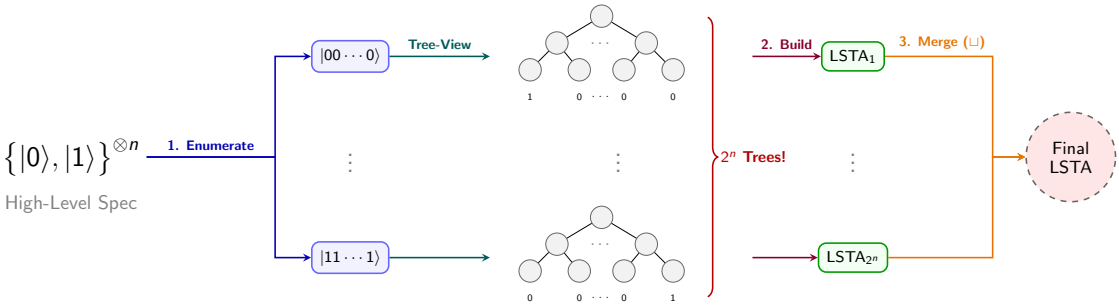
Our Language in Action: Benchmark Specifications

Benchmark	Precondition	Postcondition	Benchmark	Precondition	Postcondition
BV	$\{\ \lvert s0^n0\rangle : \lvert s\rvert = n\ \}$	$\{\ \lvert ss0\rangle : \lvert s\rvert = n\ \}$	MCToffoli ($\neg$flip, $t=0$)	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}0\rangle : i \neq 1^n\ \}$
GHZ	$\{\ \lvert i\rangle : \lvert i\rvert = n\ \}$	$\{\ \frac{1}{\sqrt{2}}\lvert 0i\rangle + \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle,\ \frac{1}{\sqrt{2}}\lvert 0i\rangle - \frac{1}{\sqrt{2}}\lvert 1\bar{i}\rangle \ :\ \lvert i\rvert = n-1\ \}$	MCToffoli ($\neg$flip, $t=1$)	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$	$\{\ \lvert i0^{n-1}1\rangle : i \neq 1^n\ \}$
Grover	$\{\ \lvert s0^n0^{n-2}0\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \lvert a_h\rvert^2>7/8}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=0$)	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$
GroverIter	$\bigcup_{\substack{\mathrm{imag}(a_h)=0,\\ \mathrm{real}(a_h)>0,\\ \mathrm{imag}(a_\ell)=0,\\ \mathrm{real}(a_\ell)>0,\\ 7a_\ell>a_h}} \{\ a_h\lvert ss0^{n-2}1\rangle + a_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	$\bigcup_{\substack{\mathrm{imag}(a'_h)=0,\\ \mathrm{imag}(a'_\ell)=0,\\ \lvert a'_h\rvert>\lvert a_h\rvert}} \{\ a'_h\lvert ss0^{n-2}1\rangle + a'_\ell \sum_{\mathbf{i}\neq\mathbf{s}} \lvert si0^{n-2}1\rangle : \lvert s\rvert = n\ \}$	MCToffoli (flip, $t=1$)	$\{\ \lvert 1^n0^{n-1}1\rangle\ \}$	$\{\ \lvert 1^n0^{n-1}0\rangle\ \}$

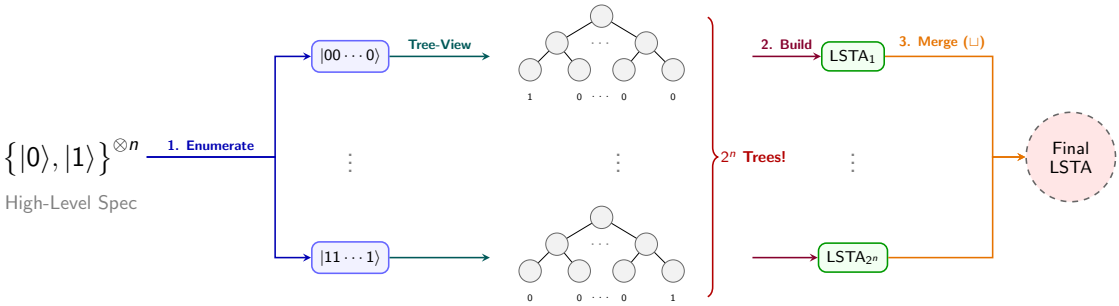
Important Observation: The specifications are now **direct mathematical descriptions** of quantum properties, which are highly intuitive for quantum researchers.

How do we translate such expressive, high-level mathematics into compact automata?

The Translation Challenge: The Naïve Exponential Wall

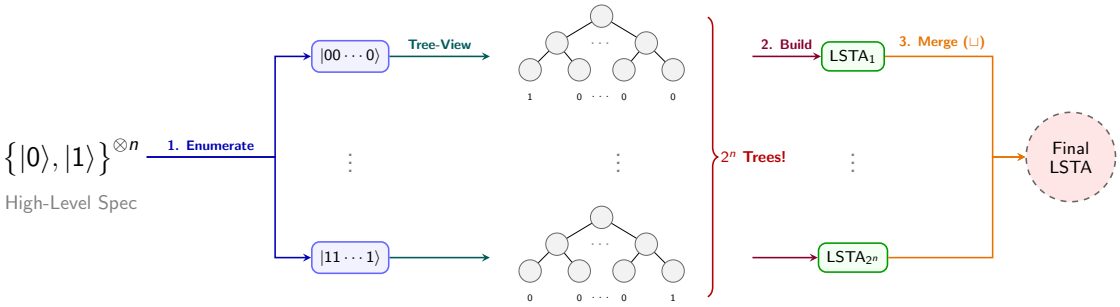


The Translation Challenge: The Naïve Exponential Wall



Result: Exponential blowup occurs in Step 1, independent of Step 2 and Step 3.

The Translation Challenge: The Naïve Exponential Wall



Result: Exponential blowup occurs in Step 1, independent of Step 2 and Step 3.

Solution: Bypass the entire enumeration process by leveraging variable relations in the specifications.

The Key Insight: Why is it Linear Time?

- We have proposed a **linear-time** algorithm in the number of qubits $\mathcal{O}(n)$ to translate specifications into compact LSTAs.

1. Set Decomposition

Variable-Level & Qubit-Level

Decompose complete set S into independent sets S_1, S_2, $\cdots$ according to some rules.

Instead of enumerating the exponential state space, we build small, localized automata for each component separately.

$$S \to S_1 \otimes S_2 \otimes \cdots \to \mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots$$

2. Efficient LSTA Combinators

Theorem: Automaton Size Bounds

Composing LSTAs is structurally efficient. The operations strictly bound the resulting size:

- **Union:** $|\mathcal{A} \sqcup \mathcal{B}| \leq |\mathcal{A}| + |\mathcal{B}|$
- **Tensor:** $|\mathcal{A} \otimes \mathcal{B}| \leq |\mathcal{A}| + N_{\text{leaves}}(\mathcal{A}) \cdot |\mathcal{B}|$

Note: $N_{leaves}(\mathcal{A})$ is the number of distinct amplitudes, which is typically not too large.

- The linear complexity comes from (1) breaking the global state enumeration into local components and (2) applying highly efficient LSTA combinators bottom-up.

Stage 1 in Action: Variable-Level Decomposition

How do we isolate dependencies in practice? Consider a highly complex set S_B from some postcondition Q, say $(S_A \cup S_B) \otimes S_C$.

1. The Raw Specification

$$S_B = \left\{ \begin{array}{l} \alpha_1 \sum_{\substack{|l|=1,\ |m|=2,\\ |n|=1,\ y=0}} |\underset{1}{l}\,\underset{2}{u}\,\underset{3}{p}\,\underset{4}{y}\,\underset{5}{m}\,\underset{6}{m}\,\underset{7}{n}\rangle \\ + \alpha_2 \sum_{\substack{|o|=1,\ |r|=1,\\ |s|=2,\ |t|=1,\\ |w|=2,\ o\neq r}} |\underset{1}{o}\,\underset{2}{r}\,\underset{3}{s}\,\underset{4}{t}\,\underset{5}{z}\,\underset{6}{w}\,\underset{7}{u}\rangle : \begin{array}{l} |p|=2,\\ |u|=1,\\ |z|=2,\\ p\neq z \end{array} \end{array} \right\}$$

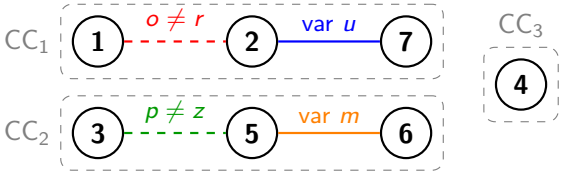
Stage 1 in Action: Variable-Level Decomposition

How do we isolate dependencies in practice? Consider a highly complex set S_B from some postcondition Q, say $(S_A \cup S_B) \otimes S_C$.

1. The Raw Specification

$$S_B = \left\{ \begin{array}{l} \alpha_1 \sum_{\substack{|l|=1,\ |m|=2,\\ |n|=1,\ y=0}} |\underset{1}{l}\,\underset{2}{u}\,\underset{3}{p}\,\underset{4}{y}\,\underset{5}{m}\,\underset{6}{m}\,\underset{7}{n}\rangle \\ + \alpha_2 \sum_{\substack{|o|=1,\ |r|=1,\\ |s|=2,\ |t|=1,\\ |w|=2,\ o\neq r}} |\underset{1}{o}\,\underset{2}{r}\,\underset{3}{s}\,\underset{4}{t}\,\underset{5}{z}\,\underset{6}{w}\,\underset{7}{u}\rangle \end{array} : \begin{array}{l} |p|=2, \\ |u|=1, \\ |z|=2, \\ p\neq z \end{array} \right\}$$

2. Dependency Analysis (The Graph)



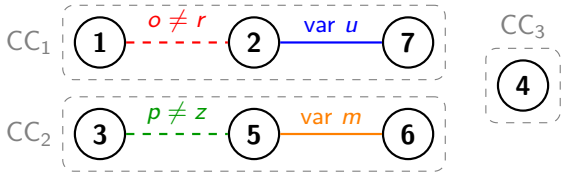
Stage 1 in Action: Variable-Level Decomposition

How do we isolate dependencies in practice? Consider a highly complex set S_B from some postcondition Q, say $(S_A \cup S_B) \otimes S_C$.

1. The Raw Specification

$$S_B = \left\{ \begin{array}{l} \alpha_1 \sum_{\substack{|l|=1,\ |m|=2,\\ |n|=1,\ y=0}} |\underset{1}{l}\,\underset{2}{u}\,\underset{3}{p}\,\underset{4}{y}\,\underset{5}{m}\,\underset{6}{m}\,\underset{7}{n}\rangle \\ + \alpha_2 \sum_{\substack{|o|=1,\ |r|=1,\\ |s|=2,\ |t|=1,\\ |w|=2,\ o\neq r}} |\underset{1}{o}\,\underset{2}{r}\,\underset{3}{s}\,\underset{4}{s}\,\underset{5}{t}\,\underset{6}{z}\,\underset{7}{w}\,\underset{7}{u}\rangle \end{array} : \begin{array}{l} |p|=2,\\ |u|=1,\\ |z|=2,\\ p\neq z \end{array} \right\}$$

2. Dependency Analysis (The Graph)



3. Component-Wise Decomposition

$$\left\{ \tau_1 \sum_{\substack{|l|=1,\\ |n|=1}} |\underset{1}{l}\,\underset{2}{u}\,\underset{7}{n}\rangle + \tau_2 \sum_{\substack{|o|=1,\\ |r|=1,\\ o\neq r}} |\underset{1}{o}\,\underset{2}{r}\,\underset{7}{u}\rangle : |u| = 1 \right\} \otimes$$

$$\left\{ \tau_1 \sum_{|m|=2} |\underset{3}{p}\,\underset{5}{m}\,\underset{6}{m}\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |\underset{3}{s}\,\underset{5}{z}\,\underset{6}{w}\rangle : \begin{array}{l} |p|=2,\\ |z|=2,\\ p\neq z \end{array} \right\} \otimes$$

$$\left\{ \tau_1 \sum_{y=0} |\underset{4}{y}\rangle + \tau_2 \sum_{|t|=1} |\underset{4}{t}\rangle \right\}$$

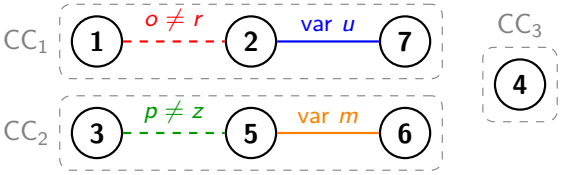
Stage 1 in Action: Variable-Level Decomposition

How do we isolate dependencies in practice? Consider a highly complex set S_B from some postcondition Q, say $(S_A \cup S_B) \otimes S_C$.

1. The Raw Specification

$$S_B = \left\{ \begin{array}{l} \alpha_1 \sum_{\substack{|l|=1,\ |m|=2,\\ |n|=1,\ y=0}} |\underset{1}{l}\ \underset{2}{u}\ \underset{3}{p}\ \underset{4}{y}\ \underset{5}{m}\ \underset{6}{m}\ \underset{7}{n}\rangle \\ + \alpha_2 \sum_{\substack{|o|=1,\ |r|=1,\\ |s|=2,\ |t|=1,\\ |w|=2,\ o\neq r}} |\underset{1}{o}\ \underset{2}{r}\ \underset{3}{s}\ \underset{4}{s}\ \underset{5}{t}\ \underset{6}{z}\ \underset{7}{w}\ \underset{7}{u}\rangle \end{array} : \begin{array}{l} |p|=2,\\ |u|=1,\\ |z|=2,\\ p\neq z \end{array} \right\}$$

2. Dependency Analysis (The Graph)



3. Component-Wise Decomposition

$$\left\{ \tau_1 \sum_{\substack{|l|=1,\\ |n|=1}} |\underset{1}{l}\ \underset{2}{u}\ \underset{7}{n}\rangle + \tau_2 \sum_{\substack{|o|=1,\\ |r|=1,\\ o\neq r}} |\underset{1}{o}\ \underset{2}{r}\ \underset{7}{u}\rangle : |u| = 1 \right\} \otimes$$

$$\left\{ \tau_1 \sum_{|m|=2} |\underset{3}{p}\ \underset{5}{m}\ \underset{6}{m}\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |\underset{3}{s}\ \underset{5}{z}\ \underset{6}{w}\rangle : \begin{array}{l} |p|=2,\\ |z|=2,\\ p\neq z \end{array} \right\} \otimes$$

$$\left\{ \tau_1 \sum_{y=0} |\underset{4}{y}\rangle + \tau_2 \sum_{|t|=1} |\underset{4}{t}\rangle \right\}$$

4. Enumeration Count

$2^{2+1+2} = 32 \longrightarrow 2^1 + 2^{2+2} + 2^0 = 19$

Significant improvement!

Stage 2 in Action: Qubit-Level Slicing & Mini-LSTAs

Now we process each isolated component independently. Let's zoom into the second component $S_{B,2} = \left\{ \tau_1 \sum_{|m|=2} |p\, m\, m\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |s\, z\, w\rangle : \begin{smallmatrix} |p|=2,\\ |z|=2,\\ p \neq z \end{smallmatrix} \right\}$, which operates on 2-qubit variables.

1. Qubit-Level Slicing

Instead of handling the full length at once, we slice the component bit-by-bit.

Length-2 Component $S_{B,2}$

$\Downarrow$

$S_{B,2}^{(1)} \otimes S_{B,2}^{(2)}$ *(Two 1-qubit slices)*

Stage 2 in Action: Qubit-Level Slicing & Mini-LSTAs

Now we process each isolated component independently. Let's zoom into the second component $S_{B,2} = \left\{ \tau_1 \sum_{|m|=2} |p\ m\ m\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |s\ z\ w\rangle : \begin{matrix} |p|=2,\\ |z|=2,\\ p \neq z \end{matrix} \right\}$, which operates on 2-qubit variables.

1. Qubit-Level Slicing

Instead of handling the full length at once, we slice the component bit-by-bit.

Length-2 Component $S_{B,2}$

$$\Downarrow$$

$$S^{(1)}_{B,2} \otimes S^{(2)}_{B,2} \quad \textit{(Two 1-qubit slices)}$$

The i-th slice encapsulates all boolean sub-constraints bound to the i-th qubits (e.g., $p_i \neq z_i$):

$$S^{(i)}_{B,2} = \left\{ \begin{matrix} \sum\limits_{m_i} f_1(p_i, z_i, m_i)\ |p_i m_i m_i\rangle \\ + \sum\limits_{s_i, w_i} f_2(p_i, z_i, s_i, w_i)\ |s_i z_i w_i\rangle \end{matrix} : p_i, z_i \right\}$$

Stage 2 in Action: Qubit-Level Slicing & Mini-LSTAs

Now we process each isolated component independently. Let's zoom into the second component $S_{B,2} = \left\{ \tau_1 \sum_{|m|=2} |p\ m\ m\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |s\ z\ w\rangle : \begin{matrix} |p|=2,\\ |z|=2,\\ p \neq z \end{matrix} \right\}$, which operates on 2-qubit variables.

1. Qubit-Level Slicing

Instead of handling the full length at once, we slice the component bit-by-bit.

Length-2 Component $S_{B,2}$

$$\Downarrow$$

$$S^{(1)}_{B,2} \otimes S^{(2)}_{B,2} \quad \textit{(Two 1-qubit slices)}$$

The i-th slice encapsulates all boolean sub-constraints bound to the i-th qubits (e.g., $p_i \neq z_i$):

$$S^{(i)}_{B,2} = \left\{ \begin{matrix} \sum\limits_{m_i} f_1(p_i, z_i, m_i)\ |p_i m_i m_i\rangle \\ + \sum\limits_{s_i, w_i} f_2(p_i, z_i, s_i, w_i)\ |s_i z_i w_i\rangle \end{matrix} : p_i, z_i \right\}$$

2. Local Enumeration & LSTA

$$S^{(i)}_{B,2} = \underbrace{\{\dots\}}_{p_i=0,\ z_i=0} \cup \underbrace{\{\dots\}}_{p_i=0,\ z_i=1} \cup \underbrace{\{\dots\}}_{p_i=1,\ z_i=0} \cup \underbrace{\{\dots\}}_{p_i=1,\ z_i=1}$$

Case 00 Case 01 Case 10 Case 11

Mini-LSTA for $S^{(i)}_{B,2}$ ⊔ (Union)

Stage 2 in Action: Qubit-Level Slicing & Mini-LSTAs

Now we process each isolated component independently. Let's zoom into the second component $S_{B,2} = \left\{ \tau_1 \sum_{|m|=2} |p\ m\ m\rangle + \tau_2 \sum_{\substack{|s|=2,\\ |w|=2}} |s\ z\ w\rangle : \begin{matrix} |p|=2,\\ |z|=2,\\ p \neq z \end{matrix} \right\}$, which operates on 2-qubit variables.

1. Qubit-Level Slicing

Instead of handling the full length at once, we slice the component bit-by-bit.

Length-2 Component $S_{B,2}$

$$\Downarrow$$

$$S^{(1)}_{B,2} \otimes S^{(2)}_{B,2} \quad \textit{(Two 1-qubit slices)}$$

The i-th slice encapsulates all boolean sub-constraints bound to the i-th qubits (e.g., $p_i \neq z_i$):

$$S^{(i)}_{B,2} = \left\{ \begin{matrix} \sum\limits_{m_i} f_1(p_i, z_i, m_i)\ |p_i m_i m_i\rangle \\ + \sum\limits_{s_i, w_i} f_2(p_i, z_i, s_i, w_i)\ |s_i z_i w_i\rangle \end{matrix} : p_i, z_i \right\}$$

2. Local Enumeration & LSTA

$$S^{(i)}_{B,2} = \underbrace{\{\dots\}}_{p_i=0,\ z_i=0} \cup \underbrace{\{\dots\}}_{p_i=0,\ z_i=1} \cup \underbrace{\{\dots\}}_{p_i=1,\ z_i=0} \cup \underbrace{\{\dots\}}_{p_i=1,\ z_i=1}$$

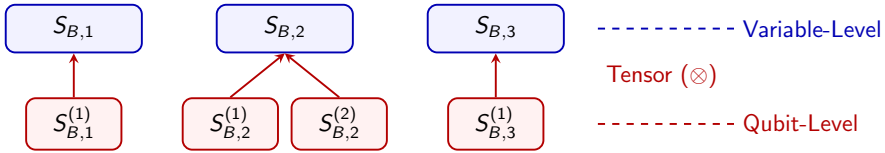
Case 00 | Case 01 | Case 10 | Case 11 → **Mini-LSTA for** $S^{(i)}_{B,2}$ ⊔ (Union)

3. The Linear Speedup

Instead of $2^{|\{p,z\}| \times 2}$ global states, we enumerate exactly $2^{|\{p,z\}|}$ states/qubit $\times$ 2 qubits. **The complexity drops from Exponential to Linear!**

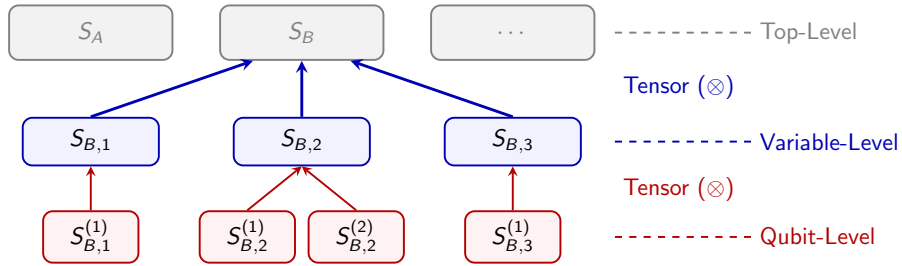
Stage 3: Final Hierarchical Assembly

The final step is to assemble all local **mini-LSTAs** bottom-up into the global specification. Crucially, we perform all combinations entirely within the **automata domain**.



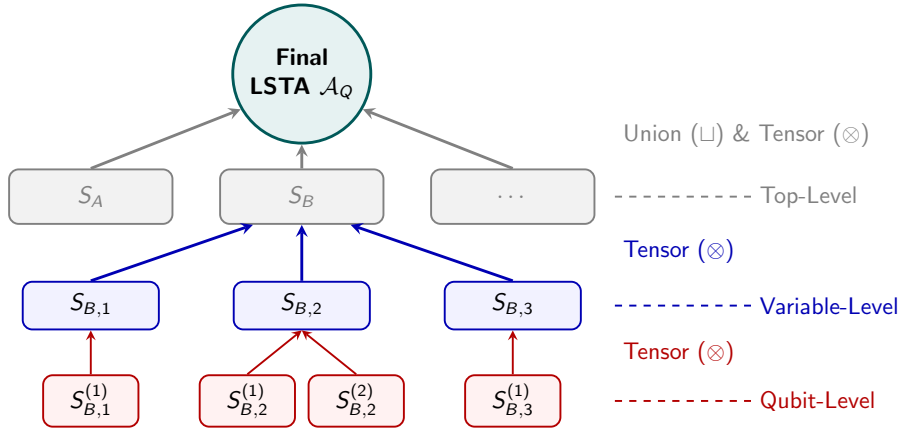
Stage 3: Final Hierarchical Assembly

The final step is to assemble all local **mini-LSTAs** bottom-up into the global specification. Crucially, we perform all combinations entirely within the **automata domain**.



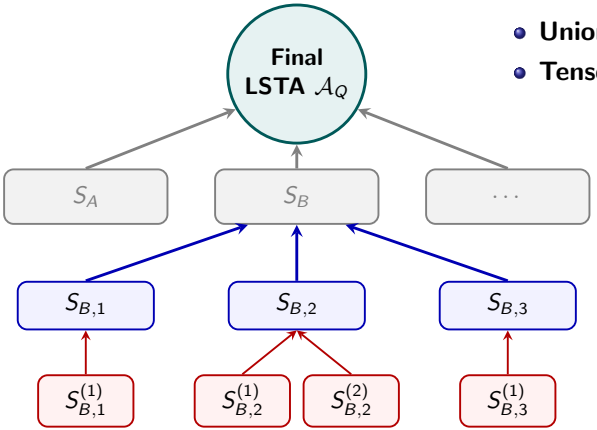
Stage 3: Final Hierarchical Assembly

The final step is to assemble all local **mini-LSTAs** bottom-up into the global specification. Crucially, we perform all combinations entirely within the **automata domain**.



Stage 3: Final Hierarchical Assembly

The final step is to assemble all local **mini-LSTAs** bottom-up into the global specification. Crucially, we perform all combinations entirely within the **automata domain**.



- **Union:** $|\mathcal{A} \sqcup \mathcal{B}| \leq |\mathcal{A}| + |\mathcal{B}|$
- **Tensor:** $|\mathcal{A} \otimes \mathcal{B}| \leq |\mathcal{A}| + N_{\text{leaves}}(\mathcal{A}) \cdot |\mathcal{B}|$

Union ($\sqcup$) & Tensor ($\otimes$)

Top-Level

Tensor ($\otimes$)

Variable-Level

Tensor ($\otimes$)

Qubit-Level

Evaluation Results

Setup: Compared against the naïve translation algorithm presented in CAV'23.
Environment: All verifications bounded by a 5-minute timeout.

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
BV	17	27	0.0s	0.0s	**0.0s**	1.5s	0.9s	**2.4s**
	19	30	0.0s	0.0s	**0.0s**	6.1s	4.3s	**10.4s**
	21	33	0.0s	0.0s	**0.0s**	25s	19.2s	**44.2s**
	23	36	0.0s	0.0s	**0.0s**	2m19s	1m35s	**3m54s**
	25	39	0.0s	0.0s	**0.0s**	$>$ 5 **minutes**		
GHZ	8	8	0.0s	0.0s	**0.0s**	0.6s	0.2s	**0.8s**
	9	9	0.0s	0.0s	**0.0s**	2.7s	1.1s	**3.8s**
	10	10	0.0s	0.0s	**0.0s**	11s	4.8s	**15.8s**
	11	11	0.0s	0.0s	**0.0s**	44s	19.1s	**1m3s**
	12	12	0.0s	0.0s	**0.0s**	$>$ 5 **minutes**		
Grover	20	544	0.0s	0.2s	**0.2s**	0.5s	2.9s	**3.4s**
	23	927	0.0s	0.5s	**0.5s**	1.9s	10.1s	**12s**
	26	1475	0.0s	1s	**1s**	7.3s	42s	**49.3s**
	29	2408	0.0s	1.9s	**1.9s**	30s	3m33s	**4m3s**
	32	3711	0.0s	3.5s	**3.5s**	$>$ 5 **minutes**		

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
Grover-Iter	20	79	0.0s	0.1s	**0.1s**	0.8s	2.7s	**3.5s**
	23	91	0.0s	0.1s	**0.1s**	2.9s	8.3s	**11.2s**
	26	103	0.0s	0.1s	**0.1s**	11s	30.7s	**41.7s**
	29	115	0.0s	0.1s	**0.1s**	50s	2m19s	**3m9s**
	32	127	0.0s	0.1s	**0.1s**	$>$ 5 **minutes**		
MC-Toffoli*	16	15	0.0s	0.0s	**0.0s**	1.2s	0.4s	**1.6s**
	18	17	0.0s	0.0s	**0.0s**	4.4s	1.6s	**6s**
	20	19	0.0s	0.0s	**0.0s**	18.1s	7.3s	**25.4s**
	22	21	0.0s	0.0s	**0.0s**	1m13s	30.4s	**1m43s**
	24	23	0.0s	0.0s	**0.0s**	$>$ 5 **minutes**		

Evaluation Results

Setup: Compared against the naïve translation algorithm presented in CAV'23.
Environment: All verifications bounded by a 5-minute timeout.

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
BV	17	27	0.0s	0.0s	**0.0s**	1.5s	0.9s	**2.4s**
	19	30	0.0s	0.0s	**0.0s**	6.1s	4.3s	**10.4s**
	21	33	0.0s	0.0s	**0.0s**	25s	19.2s	**44.2s**
	23	36	0.0s	0.0s	**0.0s**	2m19s	1m35s	**3m54s**
	25	39	0.0s	0.0s	**0.0s**	> 5 minutes		
GHZ	8	8	0.0s	0.0s	**0.0s**	0.6s	0.2s	**0.8s**
	9	9	0.0s	0.0s	**0.0s**	2.7s	1.1s	**3.8s**
	10	10	0.0s	0.0s	**0.0s**	11s	4.8s	**15.8s**
	11	11	0.0s	0.0s	**0.0s**	44s	19.1s	**1m3s**
	12	12	0.0s	0.0s	**0.0s**	> 5 minutes		
Grover	20	544	0.0s	0.2s	**0.2s**	0.5s	2.9s	**3.4s**
	23	927	0.0s	0.5s	**0.5s**	1.9s	10.1s	**12s**
	26	1475	0.0s	1s	**1s**	7.3s	42s	**49.3s**
	29	2408	0.0s	1.9s	**1.9s**	30s	3m33s	**4m3s**
	32	3711	0.0s	3.5s	**3.5s**	> 5 minutes		

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
Grover-Iter	20	79	0.0s	0.1s	**0.1s**	0.8s	2.7s	**3.5s**
	23	91	0.0s	0.1s	**0.1s**	2.9s	8.3s	**11.2s**
	26	103	0.0s	0.1s	**0.1s**	11s	30.7s	**41.7s**
	29	115	0.0s	0.1s	**0.1s**	50s	2m19s	**3m9s**
	32	127	0.0s	0.1s	**0.1s**	> 5 minutes		
MC-Toffoli*	16	15	0.0s	0.0s	**0.0s**	1.2s	0.4s	**1.6s**
	18	17	0.0s	0.0s	**0.0s**	4.4s	1.6s	**6s**
	20	19	0.0s	0.0s	**0.0s**	18.1s	7.3s	**25.4s**
	22	21	0.0s	0.0s	**0.0s**	1m13s	30.4s	**1m43s**
	24	23	0.0s	0.0s	**0.0s**	> 5 minutes		

Evaluation Results

Setup: Compared against the naïve translation algorithm presented in CAV'23.
Environment: All verifications bounded by a 5-minute timeout.

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
BV	17	27	0.0s	0.0s	**0.0s**	1.5s	0.9s	**2.4s**
	19	30	0.0s	0.0s	**0.0s**	6.1s	4.3s	**10.4s**
	21	33	0.0s	0.0s	**0.0s**	25s	19.2s	**44.2s**
	23	36	0.0s	0.0s	**0.0s**	2m19s	1m35s	**3m54s**
	25	39	0.0s	0.0s	**0.0s**	> 5 minutes		
GHZ	8	8	0.0s	0.0s	**0.0s**	0.6s	0.2s	**0.8s**
	9	9	0.0s	0.0s	**0.0s**	2.7s	1.1s	**3.8s**
	10	10	0.0s	0.0s	**0.0s**	11s	4.8s	**15.8s**
	11	11	0.0s	0.0s	**0.0s**	44s	19.1s	**1m3s**
	12	12	0.0s	0.0s	**0.0s**	> 5 minutes		
Grover	20	544	0.0s	0.2s	**0.2s**	0.5s	2.9s	**3.4s**
	23	927	0.0s	0.5s	**0.5s**	1.9s	10.1s	**12s**
	26	1475	0.0s	1s	**1s**	7.3s	42s	**49.3s**
	29	2408	0.0s	1.9s	**1.9s**	30s	3m33s	**4m3s**
	32	3711	0.0s	3.5s	**3.5s**	> 5 minutes		

	#q	#G	This Work (lin.)			Naïve Baseline (exp.)		
			trans	ver	total	trans	ver	total
Grover-Iter	20	79	0.0s	0.1s	**0.1s**	0.8s	2.7s	**3.5s**
	23	91	0.0s	0.1s	**0.1s**	2.9s	8.3s	**11.2s**
	26	103	0.0s	0.1s	**0.1s**	11s	30.7s	**41.7s**
	29	115	0.0s	0.1s	**0.1s**	50s	2m19s	**3m9s**
	32	127	0.0s	0.1s	**0.1s**	> 5 minutes		
MC-Toffoli*	16	15	0.0s	0.0s	**0.0s**	1.2s	0.4s	**1.6s**
	18	17	0.0s	0.0s	**0.0s**	4.4s	1.6s	**6s**
	20	19	0.0s	0.0s	**0.0s**	18.1s	7.3s	**25.4s**
	22	21	0.0s	0.0s	**0.0s**	1m13s	30.4s	**1m43s**
	24	23	0.0s	0.0s	**0.0s**	> 5 minutes		

- These results convince us to urgently integrate this work into AutoQ to boost efficiency.

Summary: What We Achieved

A Practical Specification Language for Automatic Quantum Program Verification

1. **The Framework (Backend)**
   - We built upon the Hoare-style framework and our AutoQ paradigm, which utilizes **LSTAs** to fully automate the verification process.
2. **The Usability Leap (Frontend)**
   - We eliminated the usability bottleneck of manual automata construction by introducing a **high-level, set-based specification language** that mirrors mathematical intuition.
3. **The Algorithmic Breakthrough (Middle-end)**
   - Our co-designed algorithm translates specs directly into LSTAs in **Linear Time** $\mathcal{O}(n)$.
   - We successfully bypassed the $\mathcal{O}(2^n)$ exponential wall.

The Result: Push-button verification of large-scale quantum circuits is now practical and accessible!

Thank You!

Q & A