

Verifying Repeat-until-Success Protocols using Automata

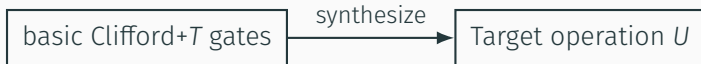
Jyun-Ao Lin

October 30, 2026, OOPSLA 2026

National Taipei University of Technology

Joint work with Yu-Fang Chen, Jakub Havlík,
Ondřej Lengál, Fang-Yi Lo, Wei-Lun Tsai, and You-Jie Wu

Repeat-Until-Success (RUS) protocol for gate synthesis



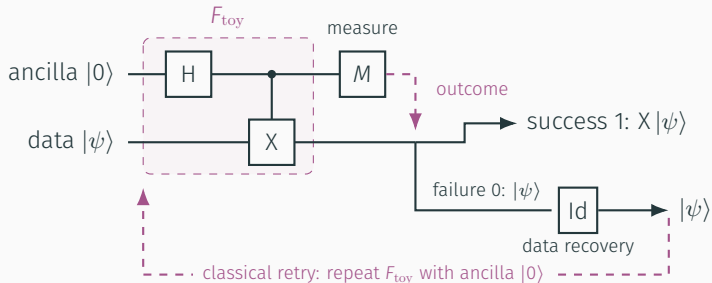
Aim: Use Clifford+ T or other basic gate sets to synthesize the desired operation U .

For suitable targets, **RUS** can reduce the expected T -gate cost.

RUS implements certain unitaries exactly and helps approximate rotations.

Repeat-Until-Success (RUS) protocol for gate synthesis

A toy attempt: target X , failure Id



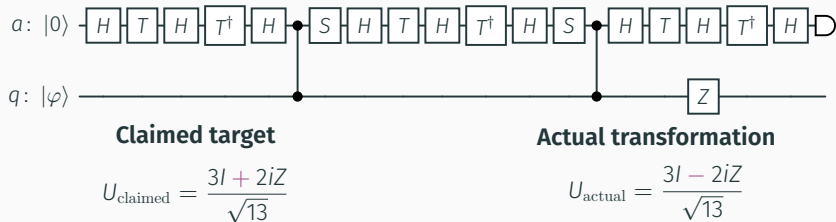
Normalized branch data: success 1 gives $X|\psi\rangle$; failure 0 gives $|\psi\rangle$.

Every **terminating** execution must give the intended output for its original input.

Paetznick-Svore, QIC 2014.

We found errors in published RUS protocols

Paetznick–Svore, 2014, Fig. 10(a)



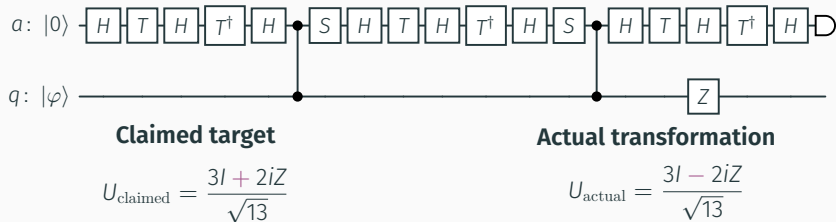
These differ by more than a global phase.

A second target (Fig. 10(c)) has the same sign error:

$$\frac{5I + 2iZ}{\sqrt{29}} \rightarrow \frac{5I - 2iZ}{\sqrt{29}}.$$

We found errors in published RUS protocols

Paetznick–Svore, 2014, Fig. 10(a)



These differ by more than a global phase.

A second target (Fig. 10(c)) has the same sign error:

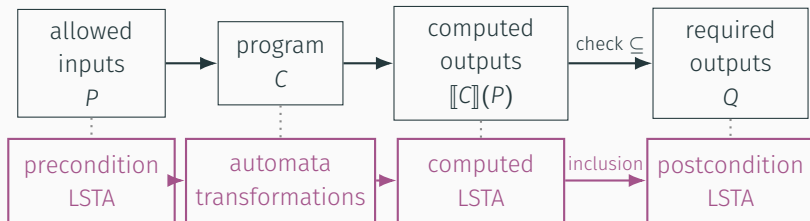
$$\frac{5I + 2iZ}{\sqrt{29}} \rightarrow \frac{5I - 2iZ}{\sqrt{29}}.$$

A wrong operation can be reused as a supposedly correct building block.

AutoQ's framework

Hoare style verification:

$$\{P\} C \{Q\}$$



Gates and measurements describe C . Loops use invariant reasoning.
AutoQ 2.0¹ already supported measurements, loops, and RUS examples.

¹Chen et al., AutoQ 2.0, TACAS 2025.

AutoQ's framework

One state

$|00\rangle$ has amplitudes $(1, 0, 0, 0)$.



Left edge: 0 right edge: 1
Leaves follow 00, 01, 10, 11.

One automaton, two trees

Structure for $\{|00\rangle, |01\rangle\}$:

$$r \longrightarrow x_1(q_1, q_0)$$

$$q_1 \longrightarrow x_2(q_3, q_5) \text{ or } x_2(q_5, q_3)$$

$$q_0 \longrightarrow x_2(q_5, q_5)$$

$$q_3 \longrightarrow 1, \quad q_5 \longrightarrow 0$$

The two runs reuse q_0 , q_3 , and q_5 .
The right subtree is always zero.

AutoQ's framework

One state

$|00\rangle$ has amplitudes $(1, 0, 0, 0)$.



Left edge: 0 right edge: 1
Leaves follow 00, 01, 10, 11.

One automaton, two trees

Structure for $\{|00\rangle, |01\rangle\}$:

$$r \longrightarrow x_1(q_1, q_0)$$

$$q_1 \longrightarrow x_2(q_3, q_5) \text{ or } x_2(q_5, q_3)$$

$$q_0 \longrightarrow x_2(q_5, q_5)$$

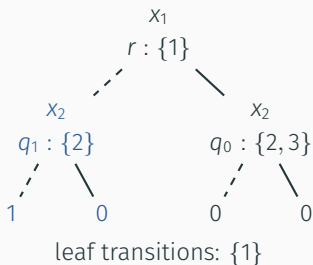
$$q_3 \longrightarrow 1, \quad q_5 \longrightarrow 0$$

The two runs reuse q_0 , q_3 , and q_5 .
The right subtree is always zero.

An LSTA adds choices to synchronize transitions at each level.

Choices synchronize a run at each level

A **level-synchronized tree automaton (LSTA)** labels each transition by a set.



A run selects transitions:

$$r \xrightarrow{\{1\}} x_1(q_1, q_0)$$

$$q_1 \xrightarrow{\{2\}} x_2(q_3, q_5)$$

$$q_1 \xrightarrow{\{3\}} x_2(q_5, q_3)$$

$$q_0 \xrightarrow{\{2,3\}} x_2(q_5, q_5)$$

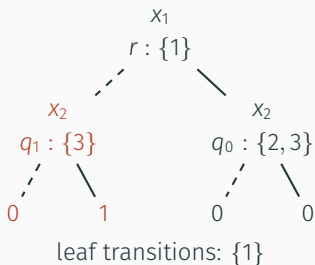
$$q_3 \xrightarrow{\{1\}} 1, \quad q_5 \xrightarrow{\{1\}} 0$$

At level 2: $\{2\} \cap \{2, 3\} = \{2\}$.

Run for $|00\rangle$: $\text{choice}(\rho_0) = (\{1\}, \{2\}, \{1\})$.

Choices synchronize a run at each level

A **level-synchronized tree automaton (LSTA)** labels each transition by a set.



A run selects transitions:

$$r \xrightarrow{\{1\}} x_1(q_1, q_0)$$

$$q_1 \xrightarrow{\{2\}} x_2(q_3, q_5)$$

$$q_1 \xrightarrow{\{3\}} x_2(q_5, q_3)$$

$$q_0 \xrightarrow{\{2,3\}} x_2(q_5, q_5)$$

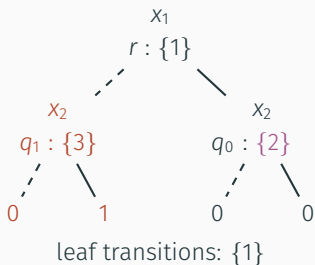
$$q_3 \xrightarrow{\{1\}} 1, \quad q_5 \xrightarrow{\{1\}} 0$$

At level 2: $\{3\} \cap \{2, 3\} = \{3\}$.

Run for |01>: $\text{choice}(\rho_1) = (\{1\}, \{3\}, \{1\})$.

Choices synchronize a run at each level

A **level-synchronized tree automaton (LSTA)** labels each transition by a set.



A run selects transitions:

$$r \xrightarrow{\{1\}} x_1(q_1, q_0)$$

$$q_1 \xrightarrow{\{2\}} x_2(q_3, q_5)$$

$$q_1 \xrightarrow{\{3\}} x_2(q_5, q_3)$$

$$q_0 \xrightarrow{\{2\}} x_2(q_5, q_5)$$

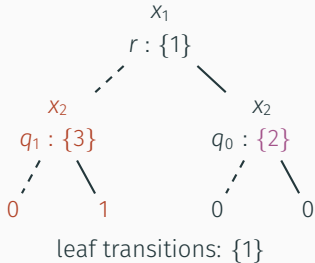
$$q_3 \xrightarrow{\{1\}} 1, \quad q_5 \xrightarrow{\{1\}} 0$$

Temporarily restrict the q_0 transition's choice set to $\{2\}$.

Candidate $|01\rangle$ **rejected**: $\{3\} \cap \{2\} = \emptyset$. No common choice at level 2.

Choices synchronize a run at each level

A **level-synchronized tree automaton (LSTA)** labels each transition by a set.



A run selects transitions:

$$r \xrightarrow{\{1\}} x_1(q_1, q_0)$$

$$q_1 \xrightarrow{\{2\}} x_2(q_3, q_5)$$

$$q_1 \xrightarrow{\{3\}} x_2(q_5, q_3)$$

$$q_0 \xrightarrow{\{2,3\}} x_2(q_5, q_5)$$

$$q_3 \xrightarrow{\{1\}} 1, \quad q_5 \xrightarrow{\{1\}} 0$$

Temporarily restrict the q_0 transition's choice set to $\{2\}$.

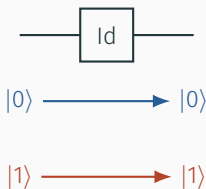
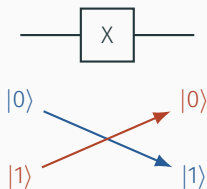
Candidate $|01\rangle$ **rejected**: $\{3\} \cap \{2\} = \emptyset$. No common choice at level 2.

LSTA reduces the complexity of basic gate transformations from exponential to quadratic in worst case

Set inclusion cannot guarantee functional correctness

Take $P = Q = \{|0\rangle, |1\rangle\}$ and ask whether $\llbracket C \rrbracket(P) \subseteq Q$.

$$\{|0\rangle, |1\rangle\} \subset \{|0\rangle, |1\rangle\}$$

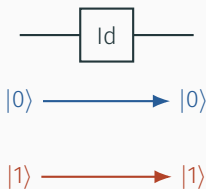
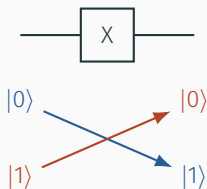


Both preserve the set. They implement different functions.

Set inclusion cannot guarantee functional correctness

Take $P = Q = \{|0\rangle, |1\rangle\}$ and ask whether $\llbracket C \rrbracket(P) \subseteq Q$.

$$\{|0\rangle, |1\rangle\} \subseteq \{|0\rangle, |1\rangle\}$$



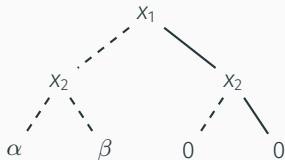
Both preserve the set. They implement different functions.

Functional correctness must preserve input-output correspondence.

Two natural approaches

Symbolic inputs in AutoQ

$$|0\rangle (\alpha |0\rangle + \beta |1\rangle)$$



α, β are shared across assertions and invariants.

Symbolic LSTA inclusion creates difficult SMT obligations.

Q: Can we replace symbolic inputs with a **finite** set of exact tests?

Simulation / finite unrolling

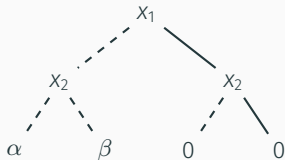
Finite executions alone do not establish the result for arbitrarily many retries.

An invariant, a summary, or an additional mathematical argument must cover the loop.

Two natural approaches

Symbolic inputs in AutoQ

$$|0\rangle (\alpha |0\rangle + \beta |1\rangle)$$



α, β are shared across assertions and invariants.

Symbolic LSTA inclusion creates difficult SMT obligations.

Q: Can we replace symbolic inputs with a **finite** set of exact tests?

e.g., can we just check $|0\rangle$ and $|1\rangle$?

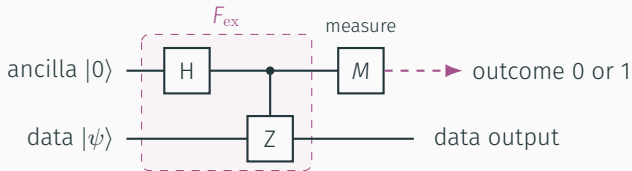
Simulation / finite unrolling

Finite executions alone do not establish the result for arbitrarily many retries.

An invariant, a summary, or an additional mathematical argument must cover the loop.

The basis tests can both pass

Example: this protocol is claimed to implement **Id** gate.



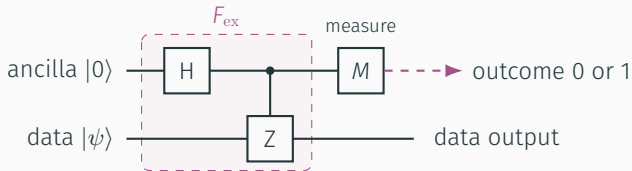
$F_{\text{ex}} = \text{CZ}(H \otimes I)$: failure 0 leaves the data unchanged.

Input

Normalized success output (1) Id check

The basis tests can both pass

Example: this protocol is claimed to implement **Id** gate.

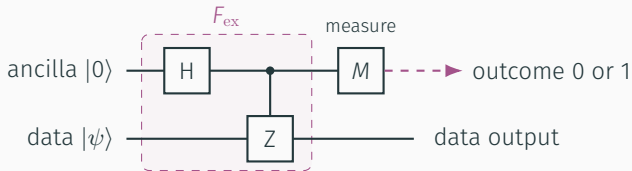


$F_{\text{ex}} = \text{CZ}(H \otimes I)$: failure 0 leaves the data unchanged.

Input	Normalized success output (1)	Id check
$ 0\rangle$	$\mapsto 0\rangle$	passes

The basis tests can both pass

Example: this protocol is claimed to implement **Id** gate.



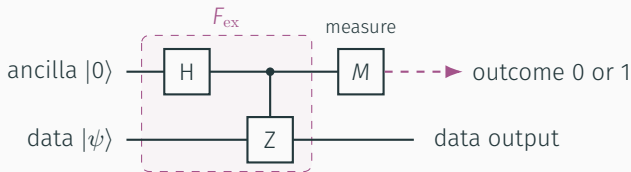
$F_{\text{ex}} = \text{CZ}(H \otimes I)$: failure 0 leaves the data unchanged.

Input	Normalized success output (1)	Id check
$ 0\rangle$	$\mapsto 0\rangle$	passes
$ 1\rangle$	$\mapsto - 1\rangle$	passes up to phase

For input $|1\rangle$, the minus sign is only a global phase.

The basis tests can both pass

Example: this protocol is claimed to implement **Id** gate.



$F_{\text{ex}} = \text{CZ}(H \otimes I)$: failure 0 leaves the data unchanged.

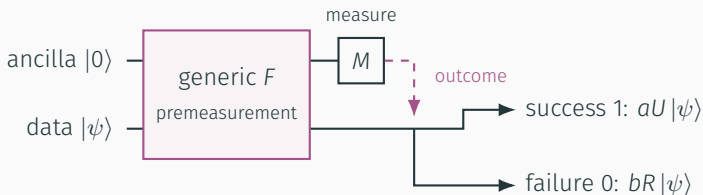
Input	Normalized success output (1)	Id check
$ 0\rangle$	$\mapsto 0\rangle$	passes
$ 1\rangle$	$\mapsto - 1\rangle$	passes up to phase
$ +\rangle = (0\rangle + 1\rangle)/\sqrt{2}$	$\mapsto -\rangle = (0\rangle - 1\rangle)/\sqrt{2}$	fails

For input $|1\rangle$, the minus sign is only a global phase.

For $|+\rangle$, the changed relative phase gives a different state.

Three tests certify the RUS step for every input

Required branch form for fixed U and R



Branch labels are unnormalized data vectors.
On failure, the applied recovery is $R^\dagger$.

Three tests certify the RUS step for every input

THEOREM (THREE-TEST CERTIFICATION)

Fix single-qubit unitaries U, R and a linear operator $F : \mathbb{C}^2 \otimes \mathbb{C}^2 \rightarrow \mathbb{C}^2 \otimes \mathbb{C}^2$. If for each $\varphi \in \{0, 1, +\}$ there are $a_\varphi, b_\varphi \in \mathbb{C}$ with

$$F(|0\rangle |\varphi\rangle) = a_\varphi |1\rangle U |\varphi\rangle + b_\varphi |0\rangle R |\varphi\rangle,$$

then there are **input-independent** $a, b \in \mathbb{C}$ such that

$$\boxed{\forall |\psi\rangle \in \mathbb{C}^2 : \quad F(|0\rangle |\psi\rangle) = a |1\rangle U |\psi\rangle + b |0\rangle R |\psi\rangle .}$$

Three tests certify the RUS step for every input

THEOREM (THREE-TEST CERTIFICATION)

Fix single-qubit unitaries U, R and a linear operator $F : \mathbb{C}^2 \otimes \mathbb{C}^2 \rightarrow \mathbb{C}^2 \otimes \mathbb{C}^2$. If for each $\varphi \in \{0, 1, +\}$ there are $a_\varphi, b_\varphi \in \mathbb{C}$ with

$$F(|0\rangle |\varphi\rangle) = a_\varphi |1\rangle U |\varphi\rangle + b_\varphi |0\rangle R |\varphi\rangle,$$

then there are **input-independent** $a, b \in \mathbb{C}$ such that

$$\boxed{\forall |\psi\rangle \in \mathbb{C}^2 : \quad F(|0\rangle |\psi\rangle) = a |1\rangle U |\psi\rangle + b |0\rangle R |\psi\rangle .}$$

Checking both branches on $|0\rangle$, $|1\rangle$, and $|+\rangle$ is enough for every input.

Why the superposition test is enough

Linearity connects the three tests.

$$F(|0+\rangle) = \frac{F(|00\rangle) + F(|01\rangle)}{\sqrt{2}}.$$

Compare the success and failure components separately:

$$a_+ U|+\rangle = \frac{a_0 U|0\rangle + a_1 U|1\rangle}{\sqrt{2}},$$

$$b_+ R|+\rangle = \frac{b_0 R|0\rangle + b_1 R|1\rangle}{\sqrt{2}}.$$

Why the superposition test is enough

Linearity connects the three tests.

$$F(|0+\rangle) = \frac{F(|00\rangle) + F(|01\rangle)}{\sqrt{2}}.$$

Compare the success and failure components separately:

$$a_+ U|+\rangle = \frac{a_0 U|0\rangle + a_1 U|1\rangle}{\sqrt{2}},$$

$$b_+ R|+\rangle = \frac{b_0 R|0\rangle + b_1 R|1\rangle}{\sqrt{2}}.$$

Unitarity makes each pair of images linearly independent.

$$a_0 = a_1 = a_+, \quad b_0 = b_1 = b_+.$$

The same a, b therefore work for $\alpha|0\rangle + \beta|1\rangle$.

Choice-aware checks keep the intended pairings

Apply X to the data qubit (level 2) of our original LSTA.

$$q \xrightarrow{\text{C}} x_2(q_L, q_R) \quad \mapsto \quad q \xrightarrow{\text{C}} x_2(q_R, q_L).$$

Swap the children; keep the choice set.

Rewrite every level-2 transition:

$$q_1 \xrightarrow{\{2\}} x_2(q_3, q_5) \quad \mapsto \quad q_1 \xrightarrow{\{2\}} x_2(q_5, q_3)$$

$$q_1 \xrightarrow{\{3\}} x_2(q_5, q_3) \quad \mapsto \quad q_1 \xrightarrow{\{3\}} x_2(q_3, q_5)$$

$$q_0 \xrightarrow{\{2,3\}} x_2(q_5, q_5) \quad \mapsto \quad q_0 \xrightarrow{\{2,3\}} x_2(q_5, q_5)$$

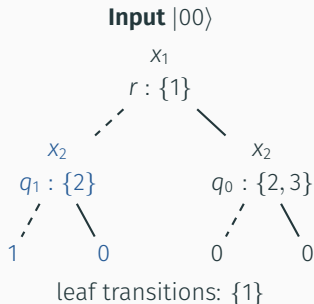
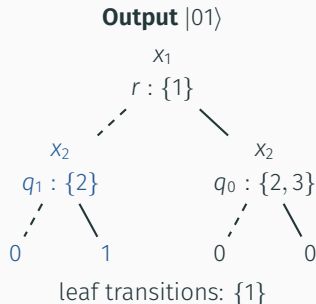
The all-zero subtree is included. Root and leaf transitions stay the same.

Exchanged subruns stay at the same depths.

Levelwise collections of choice sets, and their intersections, are unchanged.

Choice-aware checks keep the intended pairings

Follow the run for $|00\rangle$ through the X construction.



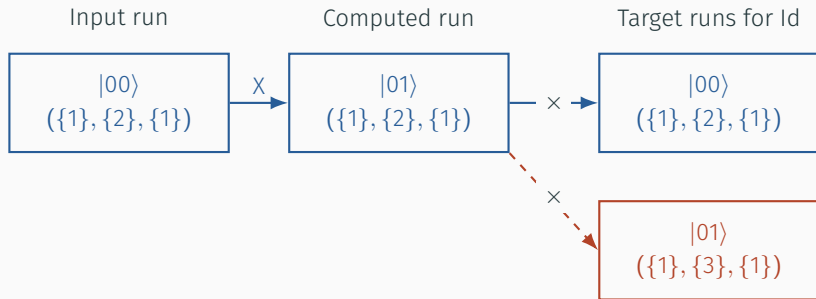
At level 2, both runs use $\{2\} \cap \{2, 3\} = \{2\}$.

$$\text{choice}(\rho_0) = \text{choice}(\rho'_0) = (\{1\}, \{2\}, \{1\}).$$

The amplitudes change from $(1, 0, 0, 0)$ to $(0, 1, 0, 0)$.

Choice-aware checks keep the intended pairings

Now check this X output against the intended operation Id.



Target $|01\rangle$: right amplitudes, incompatible choices ($\{2\} \cap \{3\} = \emptyset$).

Target $|00\rangle$: compatible choices, wrong amplitudes.

Both requirements matter. This X fails the Id specification.

Choice-aware checks keep the intended pairings

CHOICE-SEQUENCE PRESERVATION

For each accepted tree T and accepting run ρ of $\mathcal{A}$,
the inherited gate and measurement constructions yield an accepting run ρ'
of $O(\mathcal{A})$ on $O(T)$ with

$$\text{choice}(\rho') = \text{choice}(\rho).$$

The constructions are inherited; the preservation theorem is new.

General constructions may pair states and intersect choice sets.

They preserve the run's sequence, not necessarily each transition's set.

Choice-aware inclusion checks amplitudes and compatible choices.

Use a precondition that separates $|00\rangle$, $|01\rangle$, $|0+\rangle$.

An arbitrary LSTA need not separate inputs or give each tree a unique run.

Measurement leaves a whole-vector scale factor

$$a |1\rangle U |\psi\rangle + b |0\rangle R |\psi\rangle \xrightarrow{M_1^{-1}} a |1\rangle U |\psi\rangle$$

The automata keep the amplitudes without normalizing.

Consider a nonzero branch ($a \neq 0$).

Computed branch

Target representative

$$a |1\rangle U |\psi\rangle$$

$$|1\rangle U |\psi\rangle$$

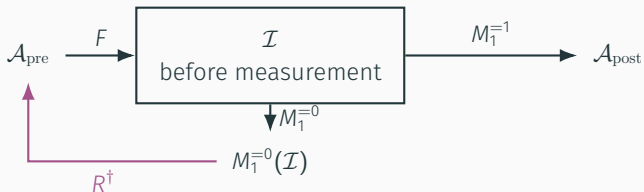
$$v \sim w \iff v = \lambda w \text{ for one } \lambda \in \mathbb{C} \setminus \{0\}.$$

Choice-aware inclusion up to scale combines this comparison
with compatible choices at every level.

Measurement transformation inherited from AutoQ 2.0.

Verifying the loop with an invariant

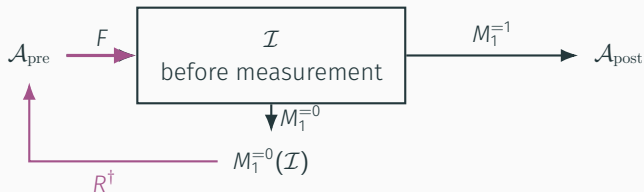
Hoare-style WHILE reasoning: initialize, preserve, establish the exit.



The guard measures and changes the state.

Verifying the loop with an invariant

Hoare-style WHILE reasoning: initialize, preserve, establish the exit.

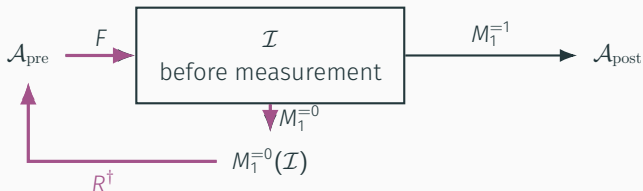


The guard measures and changes the state.

Recovery suggests $\mathcal{I} := F(\mathcal{A}_{\text{pre}})$. Initialization holds by construction.

Verifying the loop with an invariant

Hoare-style WHILE reasoning: initialize, preserve, establish the exit.



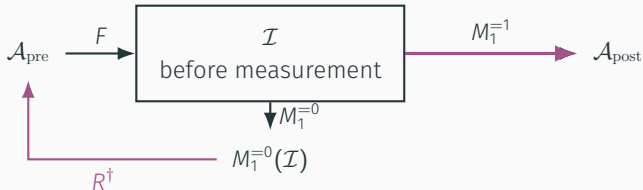
The guard measures and changes the state.

Recovery suggests $\mathcal{I} := F(\mathcal{A}_{\text{pre}})$. Initialization holds by construction.

$$\mathcal{L}(F(R^\dagger(M_1^{=0}(\mathcal{I})))) \subseteq_{\text{choice}}^{\text{uts}} \mathcal{L}(\mathcal{I}) \quad \text{retry,}$$

Verifying the loop with an invariant

Hoare-style WHILE reasoning: initialize, preserve, establish the exit.



The guard measures and changes the state.

Recovery suggests $\mathcal{I} := F(\mathcal{A}_{\text{pre}})$. Initialization holds by construction.

$$\mathcal{L}(F(R^\dagger(M_1^=0(\mathcal{I})))) \subseteq_{\text{choice}}^{\text{uts}} \mathcal{L}(\mathcal{I}) \quad \text{retry,}$$

$$\mathcal{L}(M_1^=1(\mathcal{I})) \subseteq_{\text{choice}}^{\text{uts}} \mathcal{L}(\mathcal{A}_{\text{post}}) \quad \text{exit.}$$

Evaluation: single RUS protocols

ID	Target	Gates	Result	ms	MB
V_7	$(2X + \sqrt{2}Y + Z)/\sqrt{7}$	30	OK	116	27
V_8	$(I + i\sqrt{2}X)/\sqrt{3}$	17	OK	84	27
V_9	$(I + 2iZ)/\sqrt{5}$	27	OK	89	27
$V_{10a-old}$	$(3I + 2iZ)/\sqrt{13}$	43	failed	93	25
V_{10a}	$(3I - 2iZ)/\sqrt{13}$	43	OK	97	27
V_{10b}	$(4I + iZ)/\sqrt{17}$	76	OK	104	27
$V_{10c-old}$	$(5I + 2iZ)/\sqrt{29}$	67	failed	96	27
V_{10c}	$(5I - 2iZ)/\sqrt{29}$	67	OK	95	27

All rows use 2 qubits. Minus-sign rows are the corrected targets.

The 25 MB entry is reproduced exactly from the camera-ready table; the single-protocol prose instead reports 27 MB (pairs: 26–27 MB).

Evaluation: sequential compositions

Program	Gates	Result	ms	MB
$V_7 \circ V_7$	61	OK	121	27
$V_7 \circ V_8$	48	OK	167	26
$V_8 \circ V_7$	48	OK	137	26
$V_8 \circ V_8$	35	OK	106	27
$V_8 \circ V_9$	45	OK	133	27
$V_9 \circ V_{10a}$ (corrected)	71	OK	127	27
$V_9 \circ V_{10a\text{-old}}$	71	failed	146	27
$V_{10a} \circ V_{10b}$ (corrected)	120	OK	114	27
$V_{10b} \circ V_{10c}$ (corrected)	144	OK	129	27
$V_{10b} \circ V_{10c\text{-old}}$	144	failed	154	26

Takeaways

Check the specification.

A correct proof of output-set membership can still miss the intended input-output relation.

Before adding more machinery, revisit the mathematics.

Here, elementary linear algebra reduces the RUS step's all-input requirement to three exact input checks.