Specification, Semantics, and Verification of Quantum Programs

Jennifer Paykin

University of Vermont

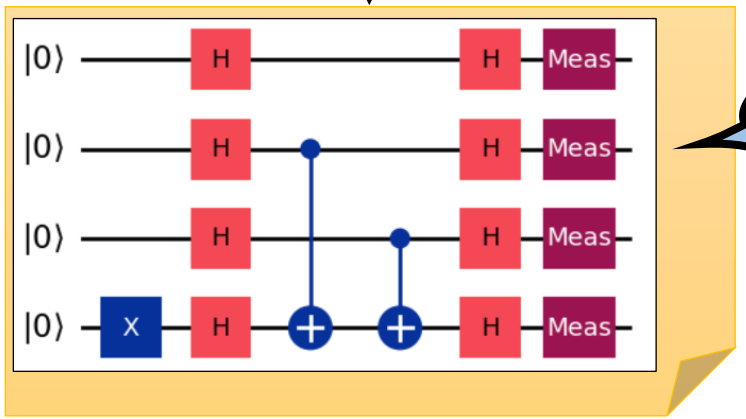
jpaykin@uvm.edu

Certified Programs and Proofs (CPP)
Jan 13, 2026

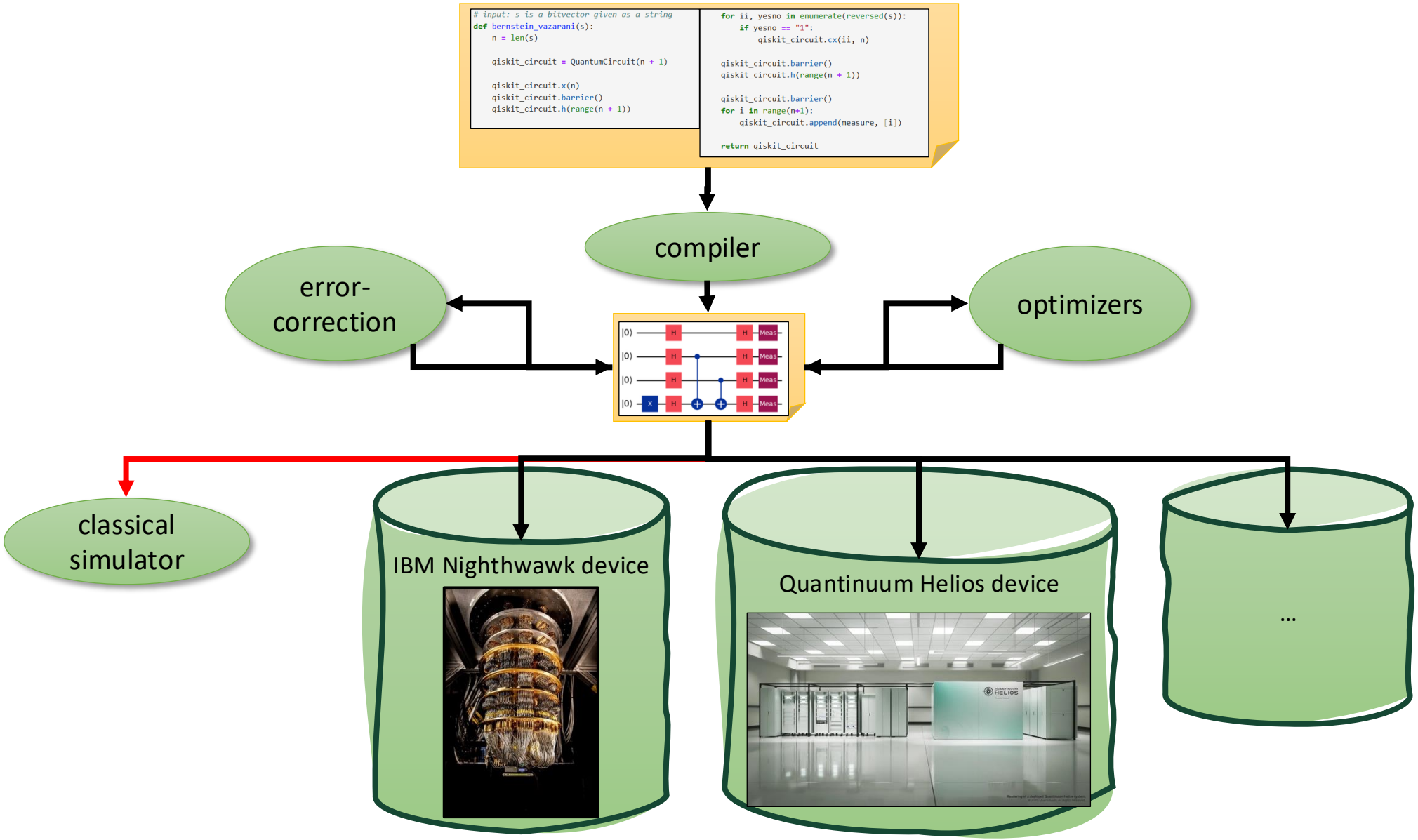
```python
# input: s is a bitvector given as a string
def bernstein_vazarani(s):
    n = len(s)

    qiskit_circuit = QuantumCircuit(n + 1)

    qiskit_circuit.x(n)
    qiskit_circuit.barrier()
    qiskit_circuit.h(range(n + 1))

    for ii, yesno in enumerate(reversed(s)):
        if yesno == "1":
            qiskit_circuit.cx(ii, n)

    qiskit_circuit.barrier()
    qiskit_circuit.h(range(n + 1))

    qiskit_circuit.barrier()
    for i in range(n+1):
        qiskit_circuit.append(measure, [i])

    return qiskit_circuit
```

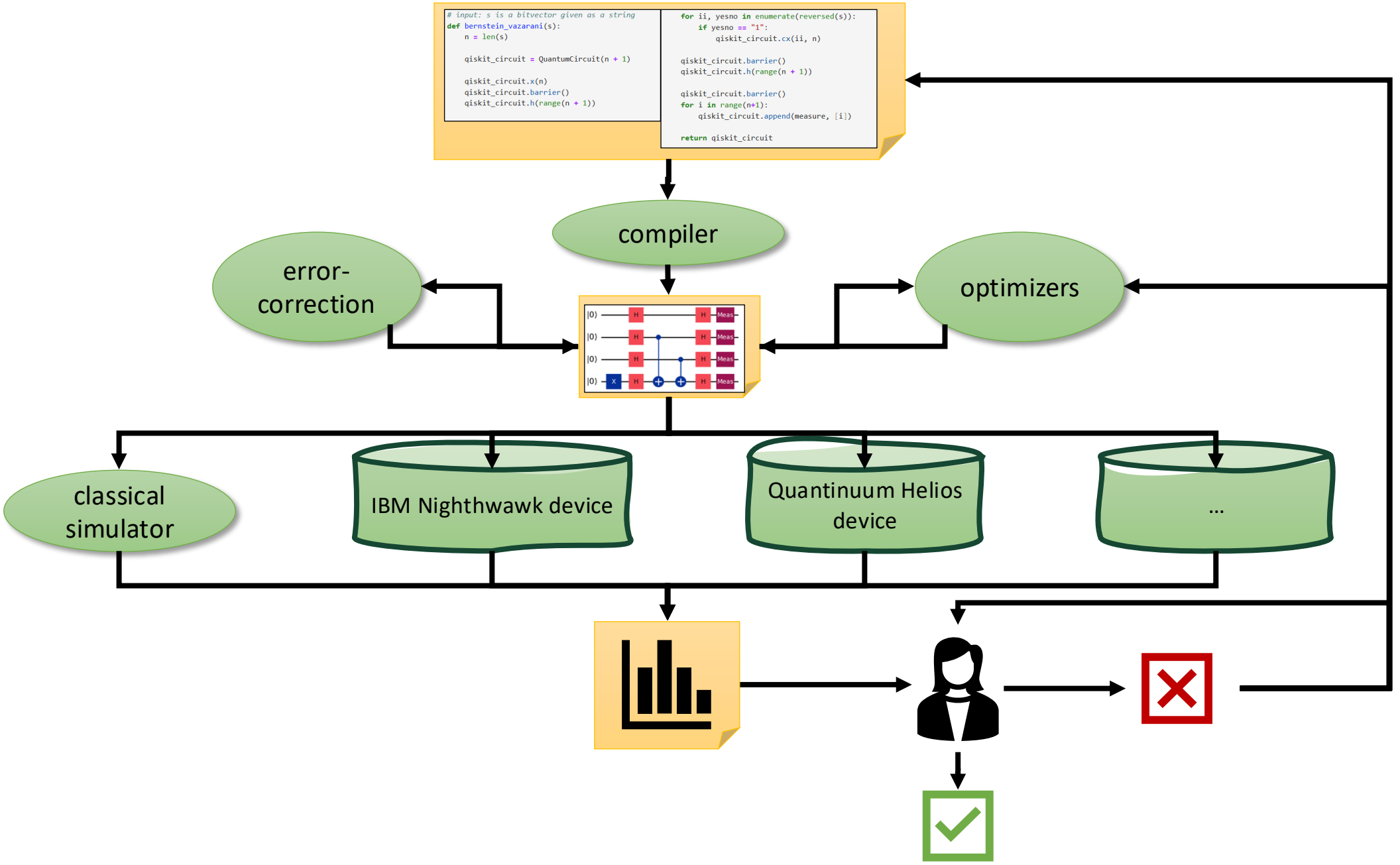
quantum DSL

compiler



quantum circuit


compiler
error-correction
optimizers
classical simulator
IBM Nighthawk device
Quantinuum Helios device
...


compiler
error-correction
optimizers
classical simulator
IBM Nighthawk device
Quantinuum Helios device
...

Outline

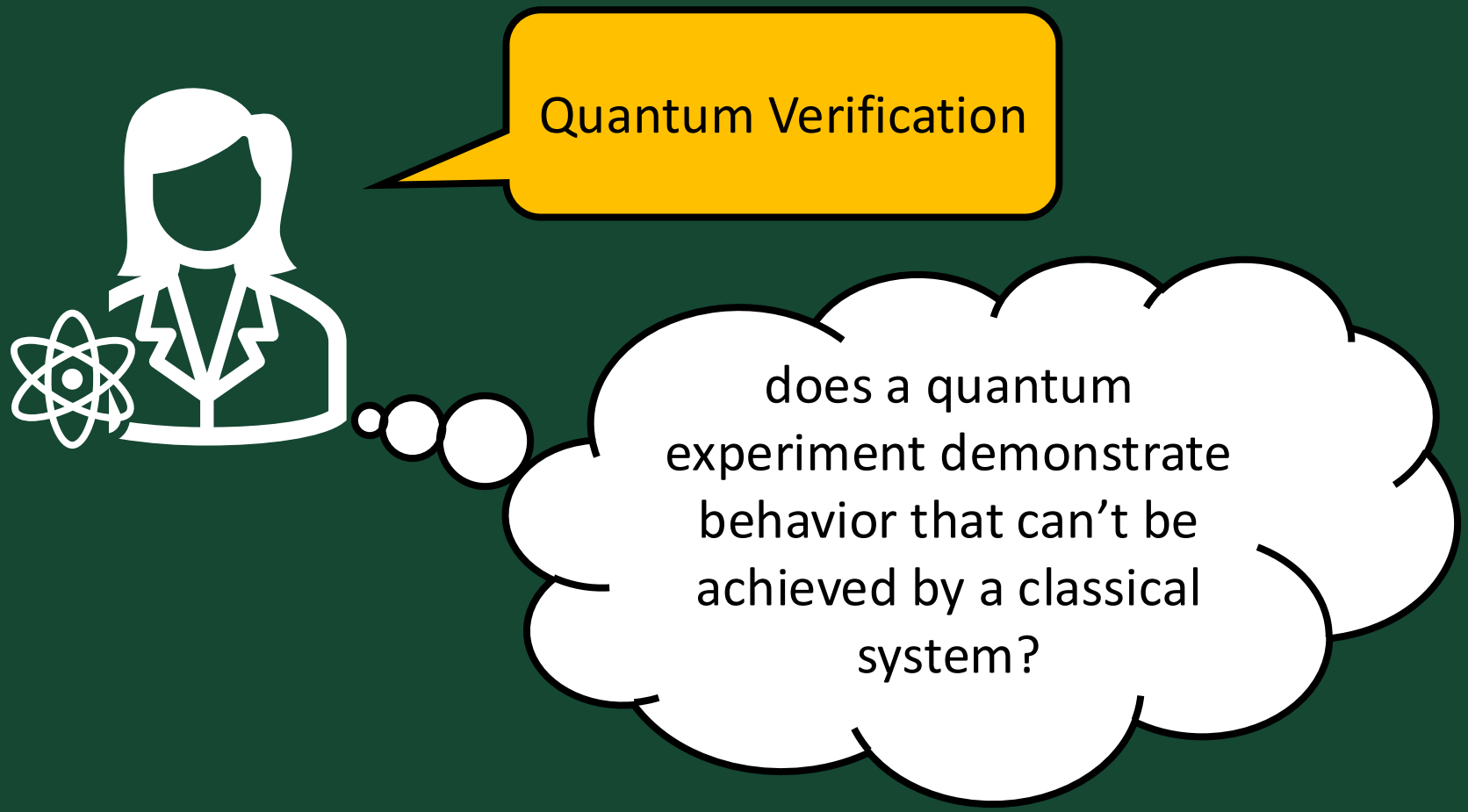
0. Vocabulary
1. Semantics of quantum circuits
2. Case study: specifying and verifying quantum programs in Rocq
3. Case study: verifying error correction and fault tolerance
4. Wrap up

Vocab Lesson

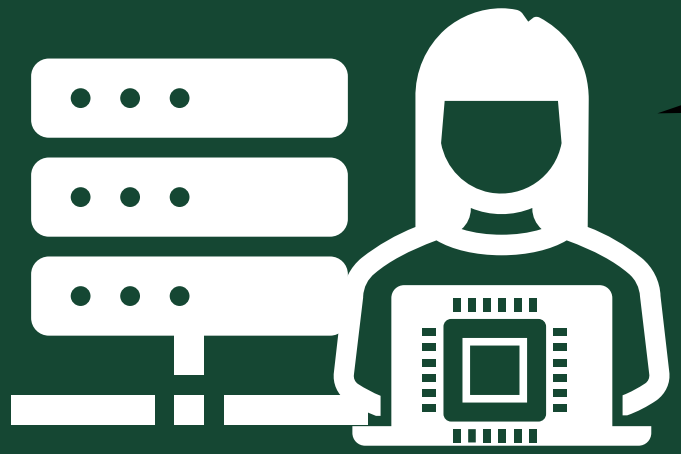
Formal Verification

machine-aided mathematical & logical methods to prove a program matches its specification

Vocab Lesson



Vocab Lesson

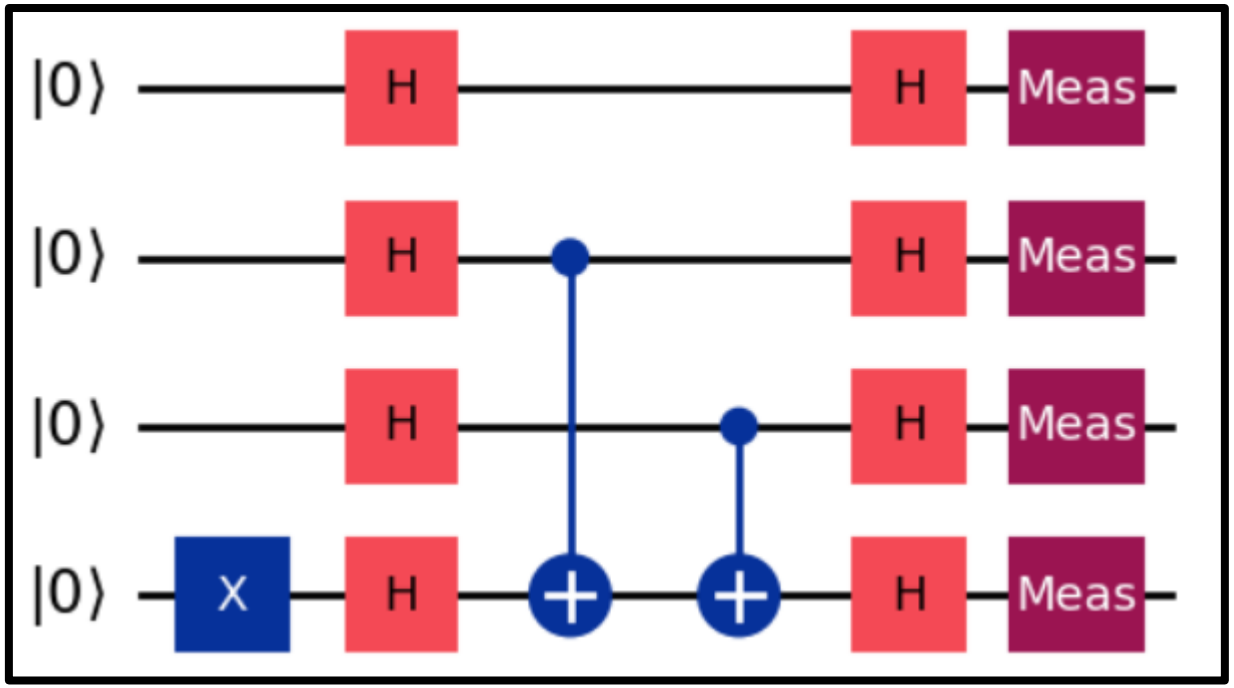


Verification & Validation

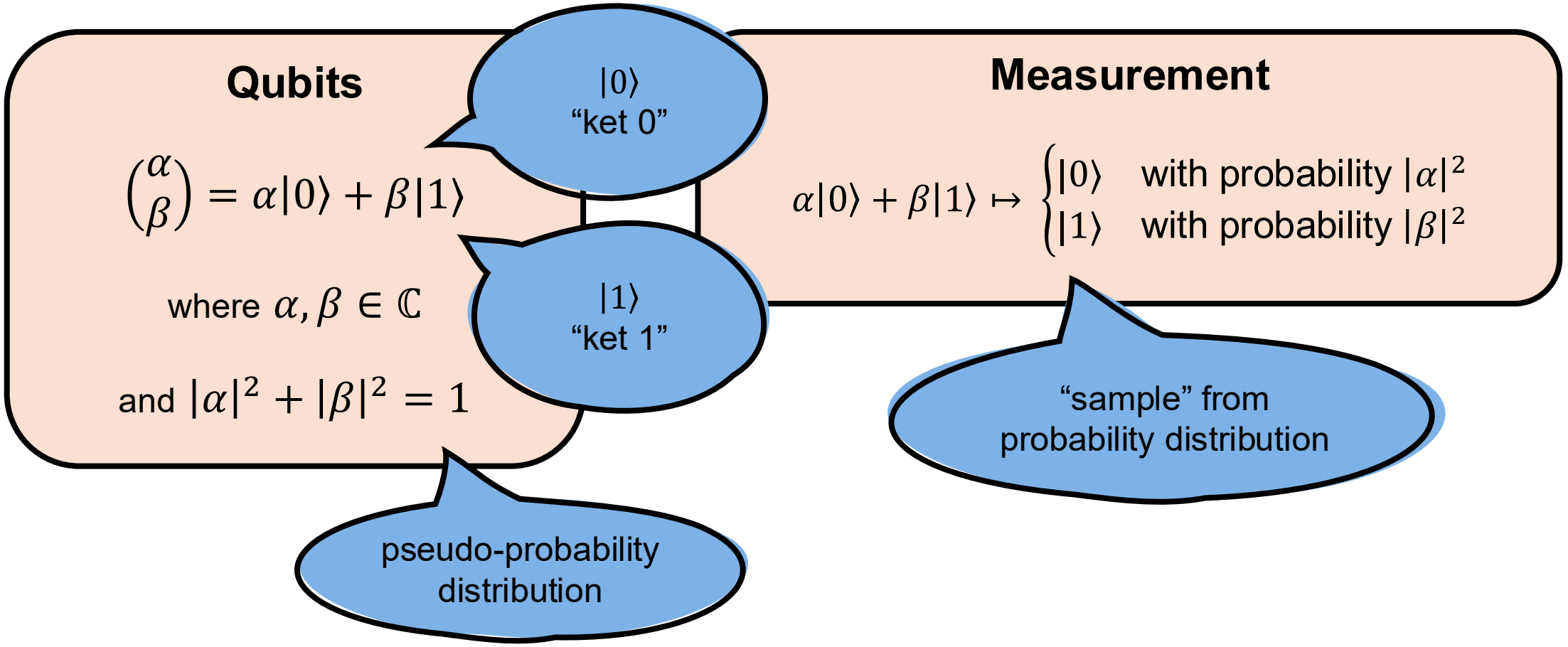
verification
=
model checking + formal verification,
OR
simulation, timing analysis, testing

Semantics of a quantum circuit

Quantum circuits



Quantum circuits



Quantum circuits

Multi-qubit systems

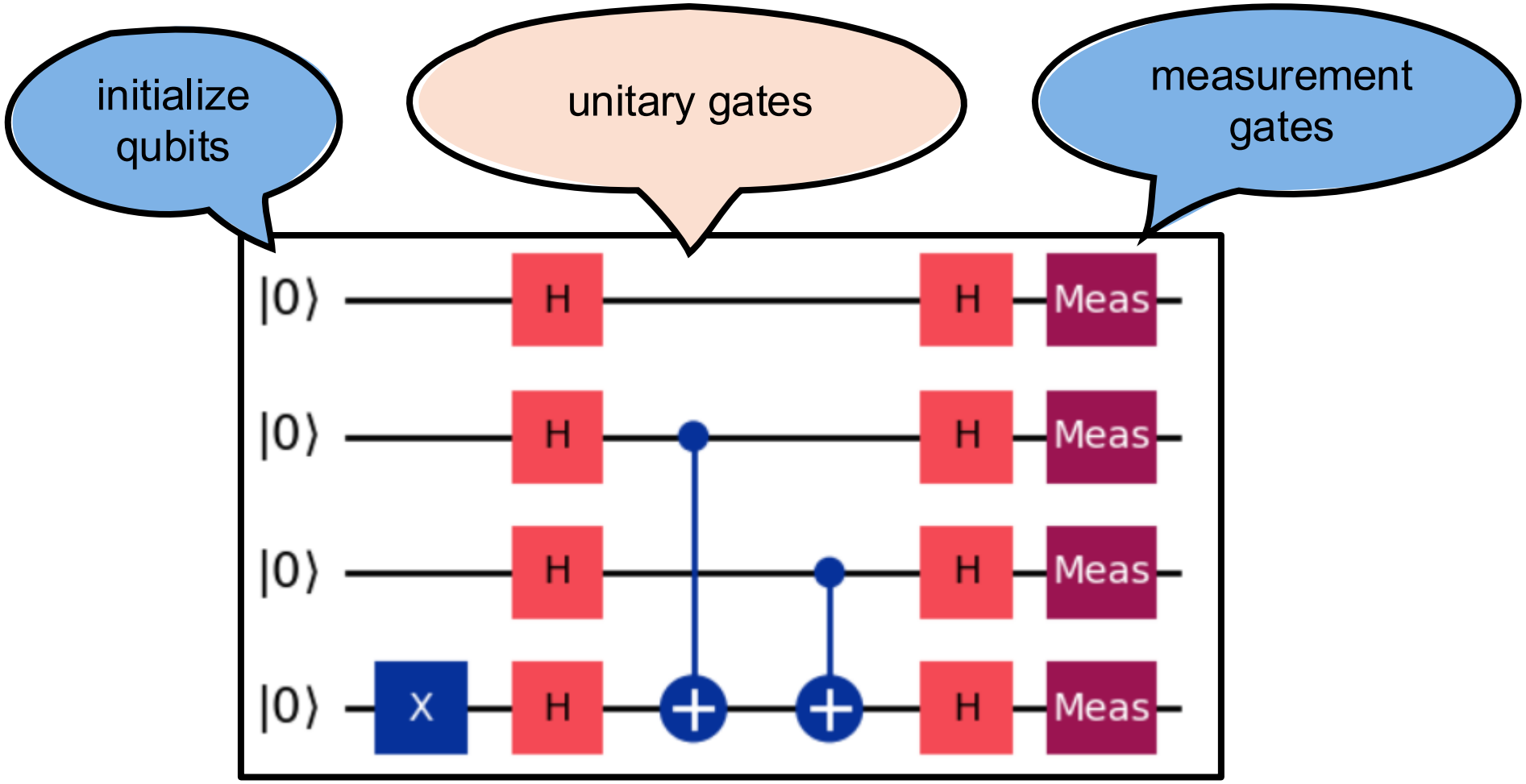
$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

where $\alpha_{ij} \in \mathbb{C}$

and $\sum|\alpha_{ij}|^2 = 1$

n qubits
=
complex vector of size 2^n

Quantum circuits



Quantum circuits

Unitary transformations

Square complex matrix $U \in \mathbb{C}^{2^n \times 2^n}$

such that $U^{-1} = U^{\dagger}$

Quantum circuits

Qubits

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$$

where $\alpha, \beta \in \mathbb{C}$

and $|\alpha|^2 + |\beta|^2 = 1$

Initialization

$$|0\rangle = \begin{pmatrix}1\\ 0\end{pmatrix} \qquad |1\rangle = \begin{pmatrix}0\\ 1\end{pmatrix}$$

Measurement

$$\alpha|0\rangle + \beta|1\rangle \mapsto \begin{cases}|0\rangle & \text{with probability } |\alpha|^2\\ |1\rangle & \text{with probability } |\beta|^2\end{cases}$$

Unitary matrices

Square complex matrix $U \in \mathbb{C}^{2^n \times 2^n}$

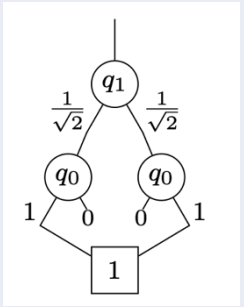
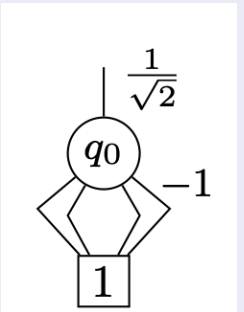
such that $U^{-1} = U^{\dagger}$

Open Problem #1: Representations of circuit semantics

Representation	$\lvert+\rangle$ state	Hadamard gate
vectors and matrices	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$
Hilbert spaces and linear transformations	$\frac{1}{\sqrt{2}}\lvert 0\rangle + \frac{1}{\sqrt{2}}\lvert 1\rangle$	$\lvert 0\rangle \mapsto \frac{1}{\sqrt{2}}\lvert 0\rangle + \frac{1}{\sqrt{2}}\lvert 1\rangle$ $\lvert 1\rangle \mapsto \frac{1}{\sqrt{2}}\lvert 0\rangle - \frac{1}{\sqrt{2}}\lvert 1\rangle$

Michael Nielsen and Isaac Chuang.
Quantum Computation and Quantum Information.
2010.

Open Problem #1: Representations of circuit semantics

Representation	$\lvert+\rangle$ state	Hadamard gate
vectors and matrices	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$
Quantum Decision Diagrams (QDDs)		

D. M. Miller and M. A. Thornton.
QMDD: A Decision Diagram Structure for Reversible and Quantum Circuits.
ISMVL 2006

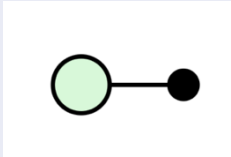
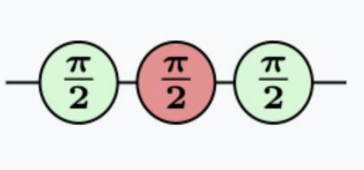
R. Wille, S. Hillmich, and L. Burgholzer.
Decision Diagrams for Quantum Computing.
In Design Automation of Quantum Computers, 2023.

Open Problem #1: Representations of circuit semantics

Representation	$\|+\rangle$ state	Hadamard gate
vectors and matrices	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$
Phase polynomials	$\frac{1}{\sqrt{2}} \sum_{x \in \mathbb{Z}_2} \|x\rangle$	$\|x\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{y \in \mathbb{Z}_2} e^{2\pi i \frac{xy}{2}} \|y\rangle$

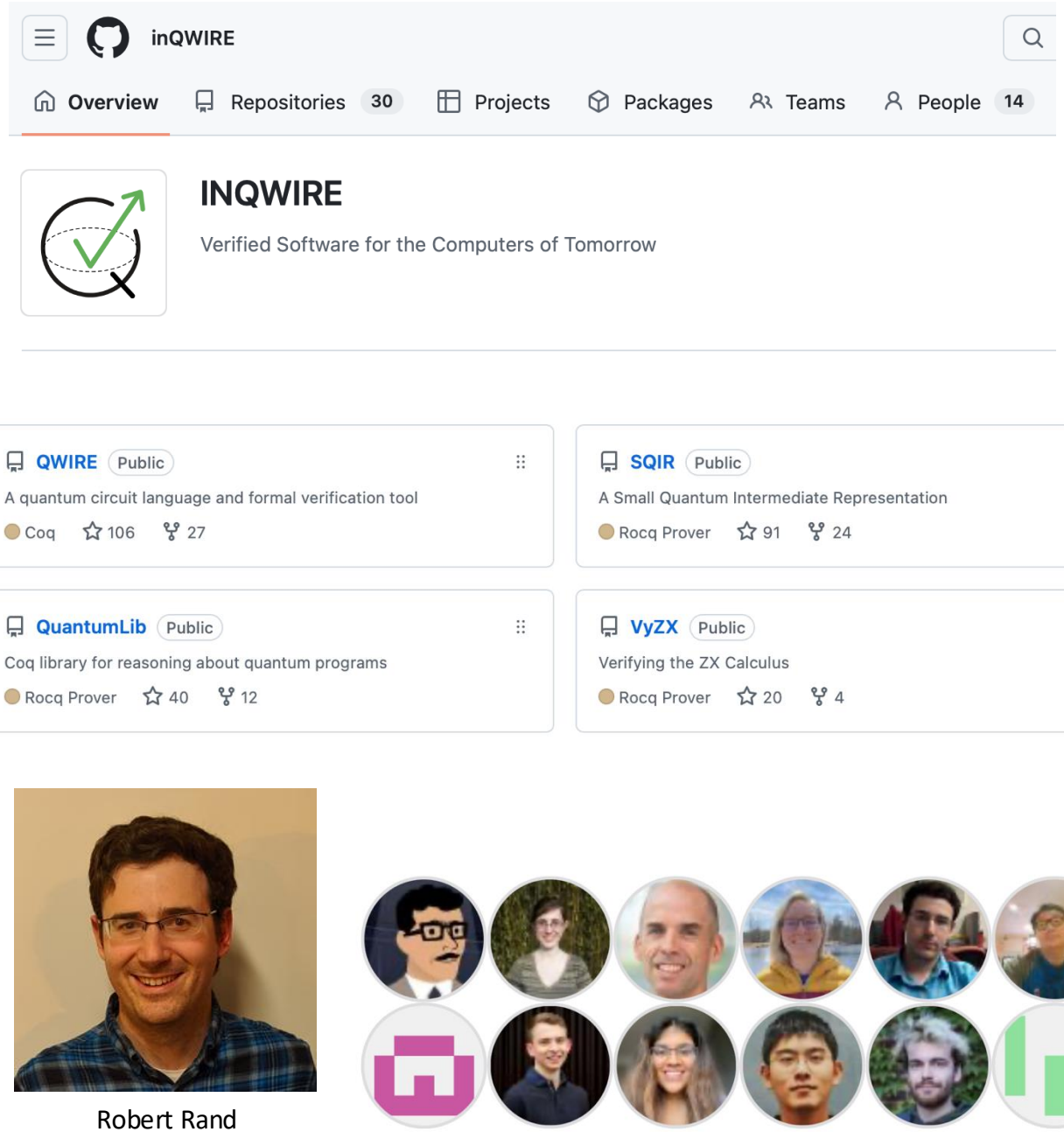
Matt Amy.
Towards large-scale functional verification of universal quantum circuits.
QPL 2017.

Open Problem #1: Representations of circuit semantics

Representation	$\|+\rangle$ state	Hadamard gate
vectors and matrices	$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$
ZX diagrams		

John van de Wetering
ZX Calculus for the Working Quantum Computer Scientist.
2020.

Case Study: Specifying and verifying quantum programs in Rocq



Robert Rand
University of Chicago

QWIRE: a core language for quantum circuits.
Jennifer Paykin, Robert Rand, and Steve Zdancewic.
POPL 2017

QWIRE practice: formal verification of quantum circuits in Coq.
Robert Rand, Jennifer Paykin, and Steve Zdancewic.
QPL 2017.

ReQWIRE: Reasoning about Reversible Quantum Circuits.
Robert Rand, Jennifer Paykin, Dong-Ho Lee, and Steve Zdancewic.
QPL 2018.

Proving Quantum Programs Correct.
Kesha Hietala, Robert Rand, Shih-Han Hung, Liyi Li, Michael Hicks.
ITP 2021

A Verified Optimizer for Quantum Circuits.
Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, Michael Hicks.
POPL 2021.

QuantumLib: A Library for Quantum Computing in Coq.
Jacob Zweifler, Kesha Hietala, Robert Rand
Coq Workshop 2022.

A Formally Certified End-to-End Implementation of Shor's Factorization Algorithm.
Yuxiang Peng, Kesha Hietala, Runzhou Tao, Liyi Li, Robert Rand, Michael Hicks, Xiaodi Wu. PNAS 2023

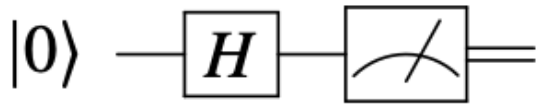
VyZX: Formal Verification of a Graphical Quantum Language.
Adrian Lehmann, Ben Caldwell, Bhakti Shah, Robert Rand.
2023.

QWIRE

quantum circuit DSL

$$\begin{cases} |0\rangle & \text{with prob. } 0.5 \\ |1\rangle & \text{with prob. } 0.5 \end{cases}$$

```
Definition coin_flip : Box One Bit :=  box_ () ⇒ meas $ _H $ init0 $ ().
```



Handling measurement probabilities

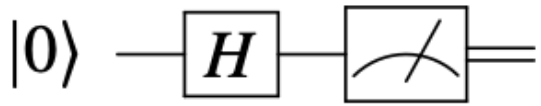
Qubits	Semantics	Measurement
pure state $\|\varphi\rangle \in \mathbb{C}^{2^n}$	$U \in \mathbb{C}^{2^n \times 2^n}$	probabilistic step $\|\varphi\rangle \to^p \|\varphi'\rangle$
mixed state *(probability distribution over pure states)* $\rho = \sum_i p_i \|\varphi_i\rangle\langle\varphi_i\| \in \mathbb{C}^{2^n \times 2^n}$	completely positive trace-preserving (CPTP) maps $\rho \mapsto \rho'$	deterministic step $\rho \to \rho'$

Michael Nielsen and Isaac Chuang.
Quantum Computation and Quantum Information.
2010.

QWIRE

```
Definition coin_flip : Box One Bit :=
        box_ () ⇒ meas $ _H $ init0 $ ().
```

$$\begin{cases} |0\rangle & \text{with prob. } 0.5 \\ |1\rangle & \text{with prob. } 0.5 \end{cases}$$



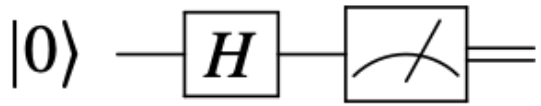
As a mixed state:

$$0.5|0\rangle\langle 0| + 0.5|1\rangle\langle 1|$$

QWIRE

$$\begin{cases} |0\rangle & \text{with prob. } 0.5 \\ |1\rangle & \text{with prob. } 0.5 \end{cases}$$

```
Definition coin_flip : Box One Bit :=
        box_ () ⇒ meas $ _H $ init0 $ ().
```



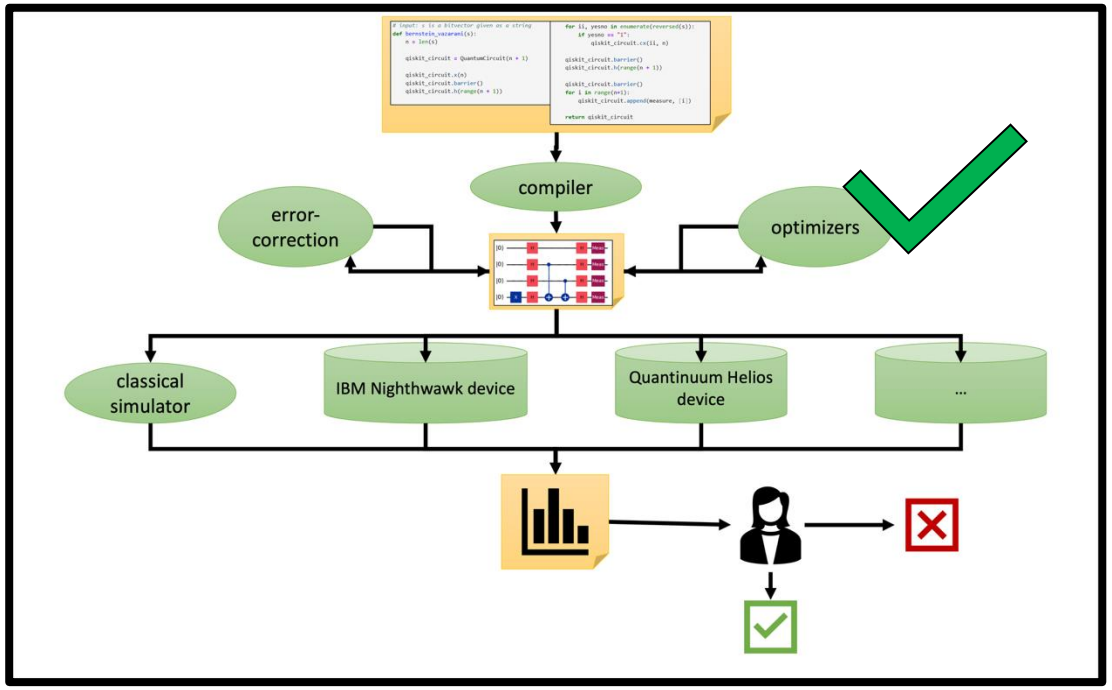

```
Lemma fair_toss : ⟦coin_flip⟧ (I 1) = 0.5 * |0⟩⟨0| + 0.5 * |1⟩⟨1|.
Proof.
  matrix_denote. Msimpl. solve_matrix.
Qed.
```

custom tactics

As a mixed state:

$$0.5|0\rangle\langle 0| + 0.5|1\rangle\langle 1|$$

VOQC



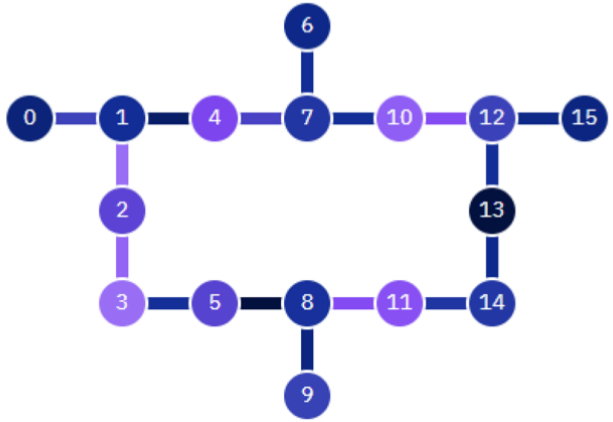
A Verified Optimizer for Quantum Circuits.
Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, Michael Hicks.
POPL 2021.

VOQC

- Verified optimizations

```
Lemma yzy_to_zyz_correct : ∀ θ1 ξ θ2 ξ1 θ ξ2,
  yzy_to_zyz θ1 ξ θ2 = (ξ1, θ, ξ2) →
  y_rotation θ2 × phase_shift ξ × y_rotation θ1
    ∝ phase_shift ξ2 × y_rotation θ × phase_shift ξ1.
```

- Verified hardware-aware passes
  - circuit mapping
  - layout
  - routing



A Verified Optimizer for Quantum Circuits.
Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, Michael Hicks.
POPL 2021.

Open Problem #2: programs vs circuits

```
# input: s is a bitvector given as a string
def bernstein_vazarani(s):
    n = len(s)

    qiskit_circuit = QuantumCircuit(n + 1)

    qiskit_circuit.x(n)
    qiskit_circuit.barrier()
    qiskit_circuit.h(range(n + 1))
```

```
    for ii, yesno in enumerate(reversed(s)):
        if yesno == "1":
            qiskit_circuit.cx(ii, n)

    qiskit_circuit.barrier()
    qiskit_circuit.h(range(n + 1))

    qiskit_circuit.barrier()
    for i in range(n+1):
        qiskit_circuit.append(measure, [i])

    return qiskit_circuit
```

quantum DSL

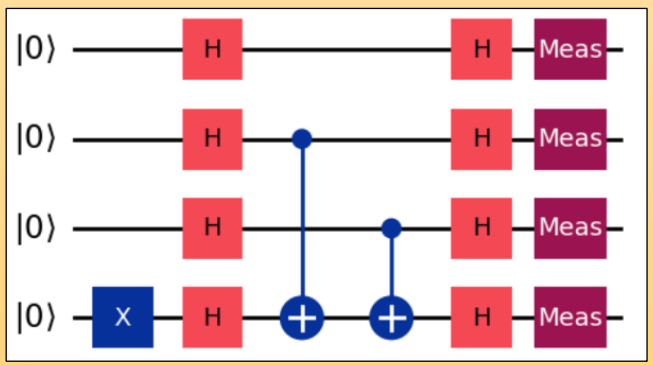
loops/classical control flow

higher-order quantum abstractions

parameterized by number of qubits, other inputs

interleaved classical-quantum computation

compiler

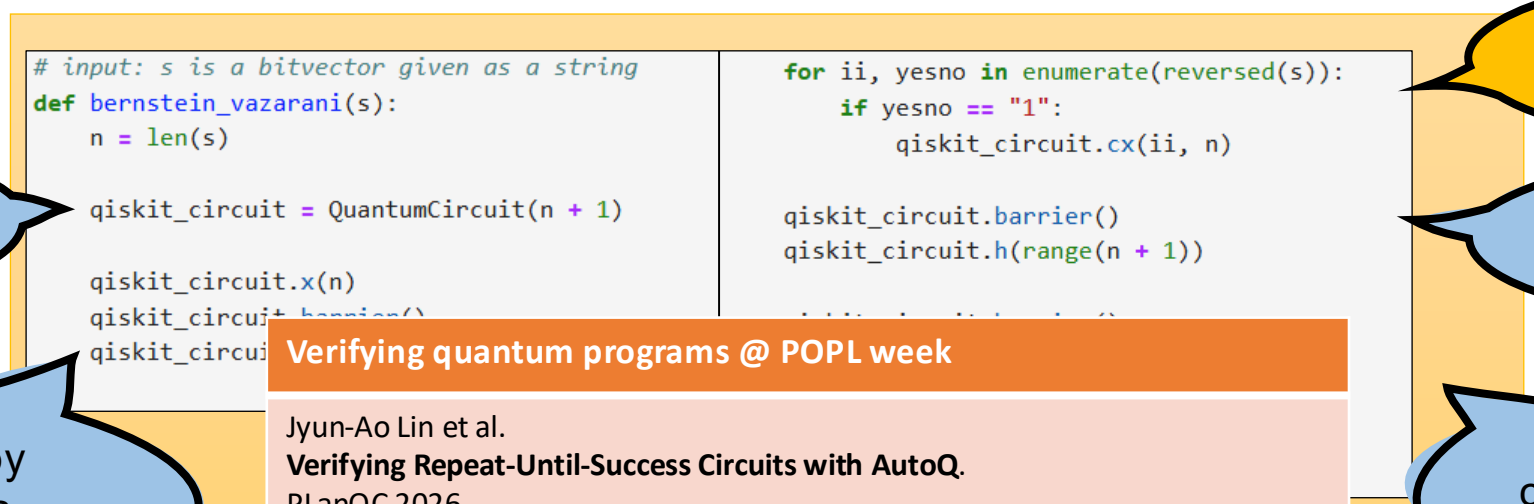


quantum circuit

Proto-Quipper family

Neil J. Ross
Algebraic and Logical Methods in Quantum Computation.
Ph.D. thesis, Dalhousie University, 2015

Open Problem #2: programs vs circuits



```
# input: s is a bitvector given as a string
def bernstein_vazarani(s):
    n = len(s)

    qiskit_circuit = QuantumCircuit(n + 1)

    qiskit_circuit.x(n)
    qiskit_circuit.barrier()
    qiskit_circuit.h(range(n + 1))
```

```
    for ii, yesno in enumerate(reversed(s)):
        if yesno == "1":
            qiskit_circuit.cx(ii, n)
```

quantum DSL

loops/classical control flow

higher-order quantum abstractions

parameterized by number of qubits, other inputs

interleaved classical-quantum computation

Verifying quantum programs @ POPL week

Jyun-Ao Lin et al.
Verifying Repeat-Until-Success Circuits with AutoQ.
PLanQC 2026.

Yingte Xu.
A Unified Assertion-Based Framework for Classical-Quantum Program Verification
PLanQC 2026.

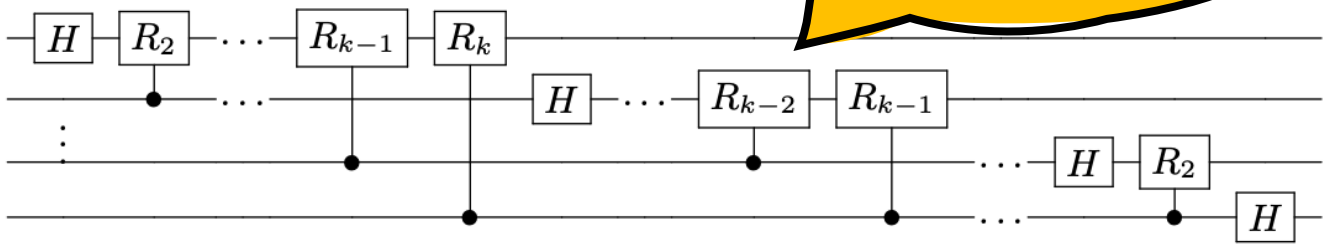
Robert Booth and Cole Comfort.
Denotational Semantics for Stabiliser Quantum Programs.
PLanQC 2026.

Parosh Aziz Abdulla et al.
Parameterized Verification of Quantum Circuits.
POPL 2026.

Ken Sakayori, Andrea Colledan, and Ugo Dal Lago.
On Circuit Description Languages, Indexed Monads, and Resource Analysis.
POPL 2026.

SQIR – Simple Quantum IR

True for any size circuit and any basis state

$QFT_k =$ 

```
Fixpoint controlled_rotations n :=
match n with
| 0 | 1 ⇒ SKIP
| S n' ⇒ controlled_rotations n' ; control n' (Rz (2π / 2n) 0)
end.
Fixpoint QFT n :
match n with
| 0 ⇒ SKIP
| S n' ⇒ H 0 ; controlled_rotations n ; map_qubits (fun q ⇒ q + 1) (QFT n')
end.
```

$$\texttt{Lemma QFT\_semantics : } \forall\ \texttt{n x, n>0} \rightarrow [\![\texttt{QFT n}]\!]_n \times |\texttt{x}\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{j=0}^{n-1} (|0\rangle + e^{2\pi i x/2^{n-j}} |1\rangle).$$

As a unitary on n qubits…

…QFT takes the basis state |x⟩…

…to the following pure state.

Proving Quantum Programs Correct.
Kesha Hietala, Robert Rand, Shih-Han Hung, Liyi Li, Michael Hicks.
ITP 2021

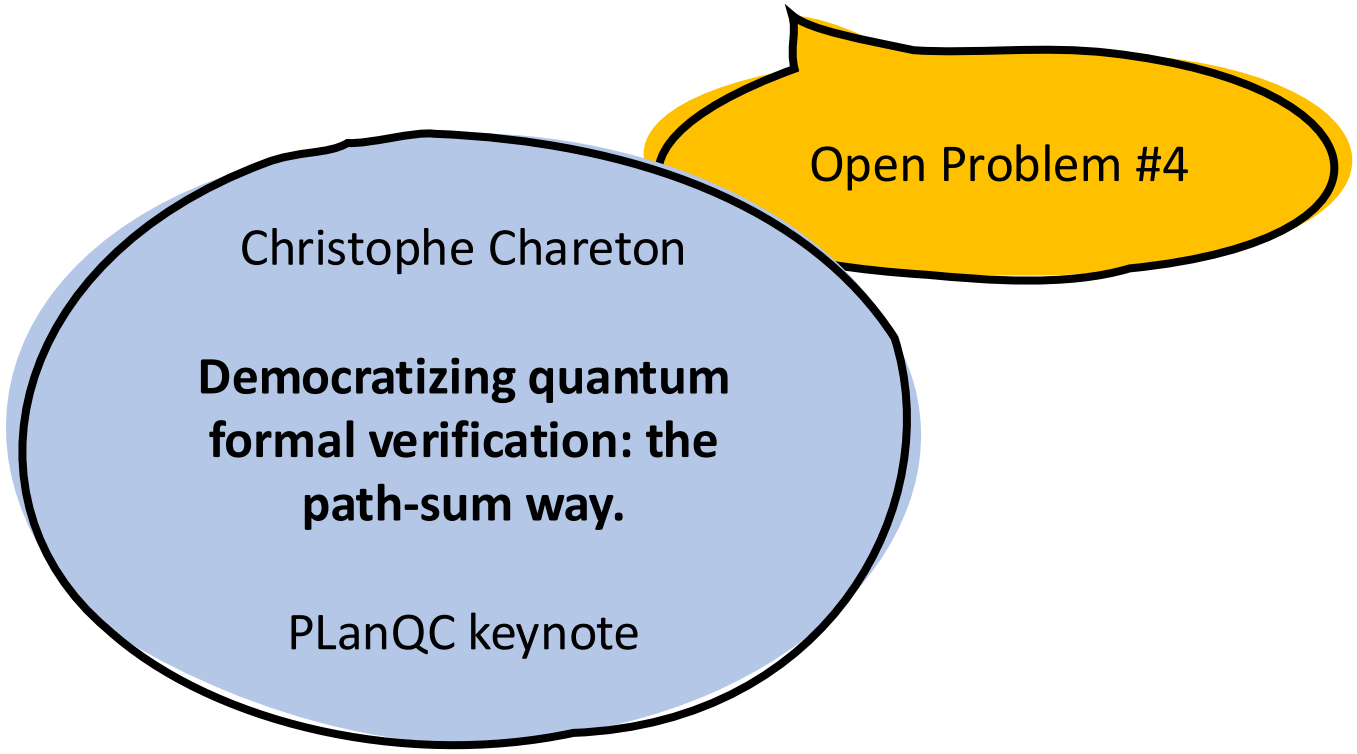
Tradeoffs

Pros	Cons
Powerful reasoning techniques	Requires expert proof developers, limited automation

Quantum Automatic Verification @ POPL Week
Xiaoguan Xu, Li Zhou, Mingsheng Ying. **VC-Qiskit: Automated–Interactive Verification of Qiskit Passes with Minimal Intrusion**
Christophe Chareton. **Democratizing quantum formal verification: the path-sum way**
Robin Adams, Jean-Philippe Bernardy, Lorenzo Perticonne, and Jeremy Pope. **A Graded Modal Type Theory for Pulse Schedules**
Xingte Xu. **A Unified Assertion-Based Framework for Classical-Quantum Program Verification**
Kayo Tei, Haruto Minshina, Naoki Yamamoto, and Kazunori Ueda. **Graph Rewriting Language as a Platform for Quantum Diagrammatic Calculi**
Marco Lewis and Benoit Valiron **Finding Photonics Circuits via δ-weakening SMT**
Parosh Aziz Abdulla et al. **Parameterized Verification of Quantum Circuits**

Tradeoffs

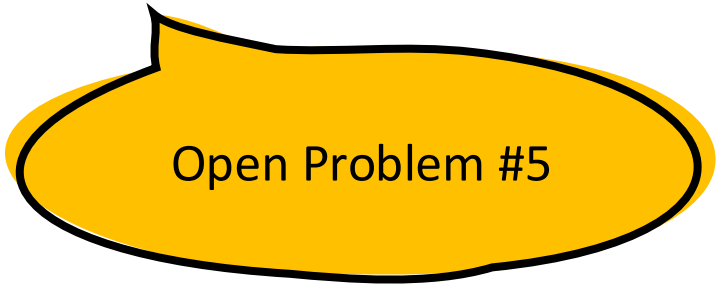
Pros	Cons
Powerful reasoning techniques	Requires expert proof developers, limited automation
Statement of equivalence conceptually clear	Other quantum specifications unintuitive



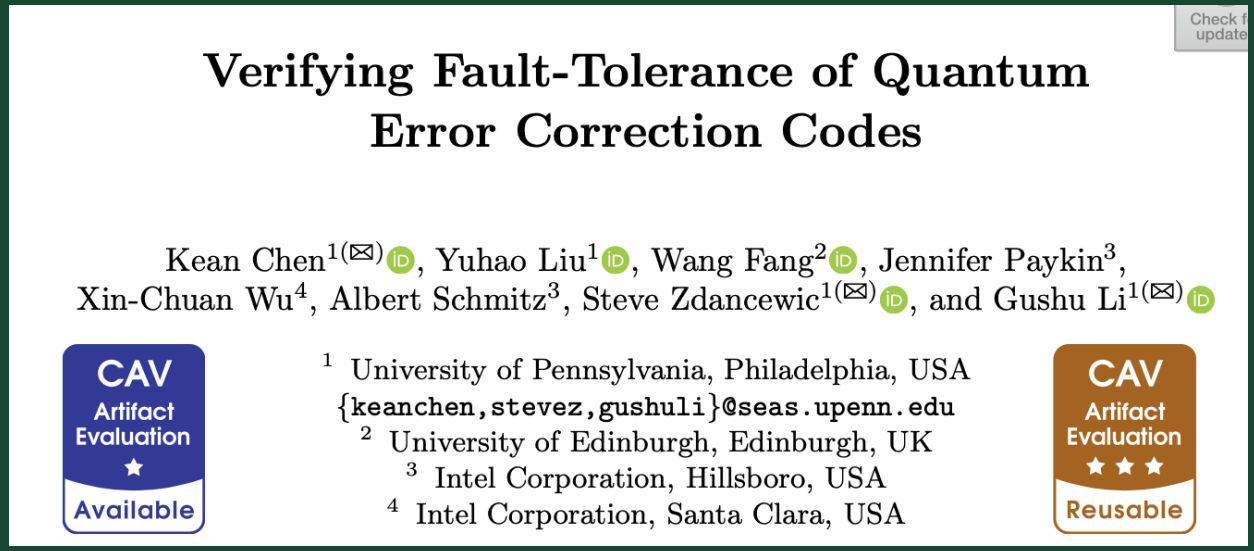
Tradeoffs

Pros	Cons
Powerful reasoning techniques	Requires expert proof developers, limited automation
Statement of equivalence conceptually clear	Other quantum specifications unintuitive
Can reason about complex properties e.g. ancillae and oracles	Not straightforward to reason about errors, probabilistic algorithms

Quantum Errors @ POPL Week
Anurudh Peduri, Jam Kabeer, Ali Khan, Gilles Barthe, Michael Walter **Traq: Estimating the Quantum Cost of Classical Programs**
Robin Adams, Jean-Philippe Bernardy, Lorenzo Perticonne, and Jeremy Pope. **A Graded Modal Type Theory for Pulse Schedules**
Jane Moore, Michael Hart, and John McAllister. **Efficient Parallel Compilation and Profiling of Quantum Circuits at Large Scales**
Yu-Hsuan Wu, Yue Shi, Junyi Liu, and Yuxiang Peng. **A Pulse-Level DSL for Real-Time Quantum Control with Hardware Compilation and Emulation**



Case Study: Verifying Error Correction and Fault Tolerance



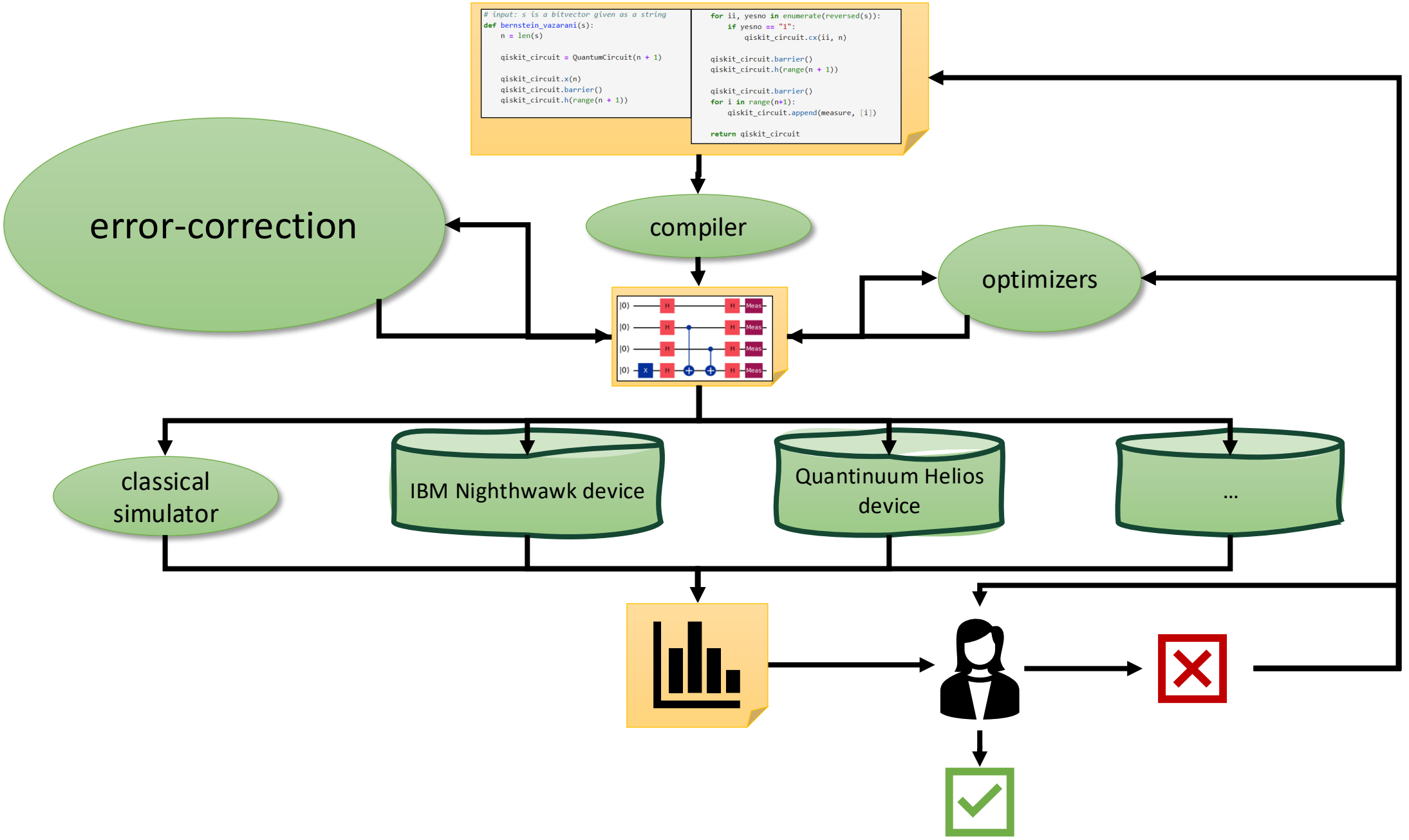
Verifying Fault-Tolerance of Quantum Error Correction Codes

Kean Chen[1(✉)], Yuhao Liu[1], Wang Fang[2], Jennifer Paykin[3], Xin-Chuan Wu[4], Albert Schmitz[3], Steve Zdancewic[1(✉)], and Gushu Li[1(✉)]

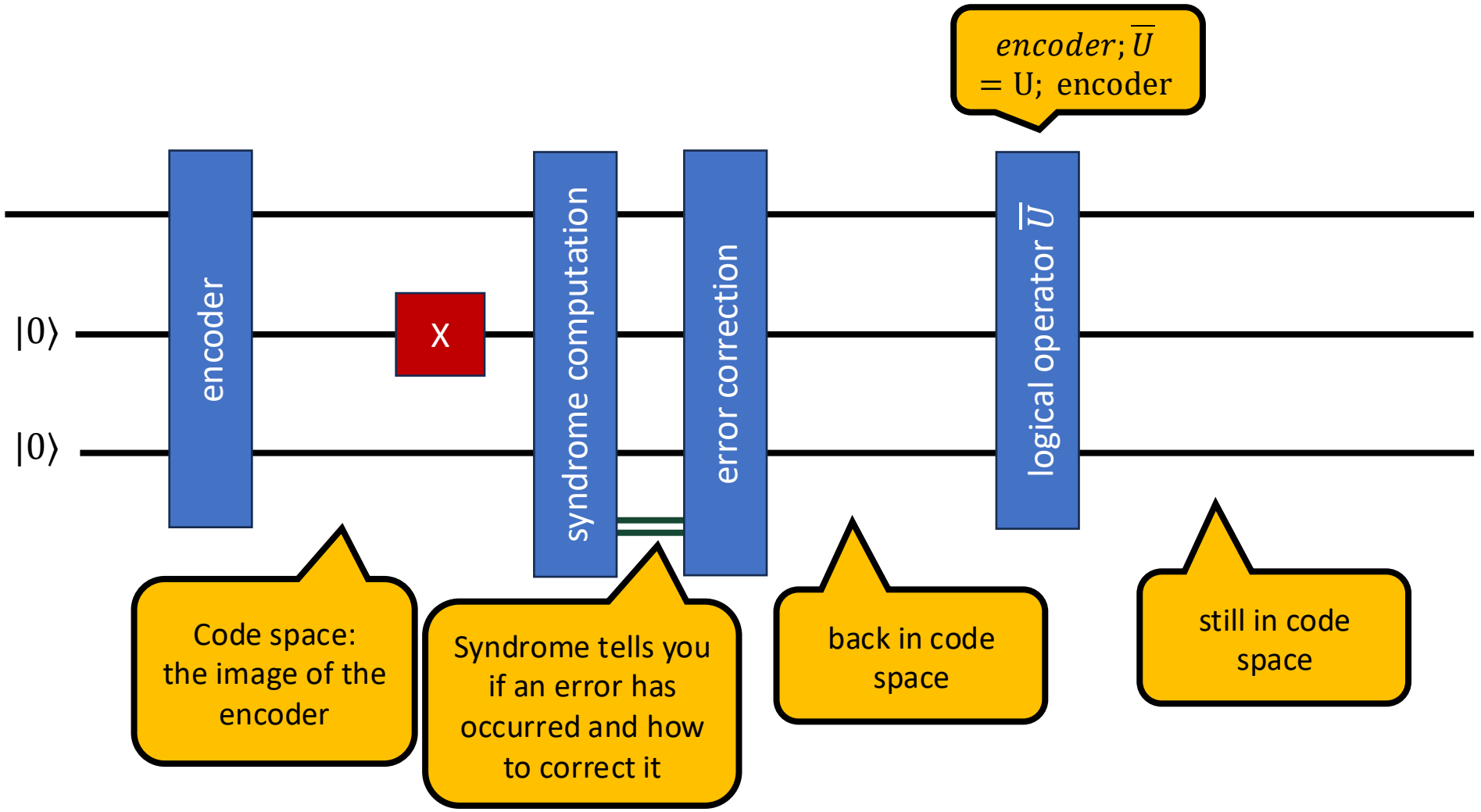
[1] University of Pennsylvania, Philadelphia, USA
{keanchen,stevez,gushuli}@seas.upenn.edu
[2] University of Edinburgh, Edinburgh, UK
[3] Intel Corporation, Hillsboro, USA
[4] Intel Corporation, Santa Clara, USA

CAV Artifact Evaluation ★ Available

CAV Artifact Evaluation ★★★ Reusable


error-correction
compiler
optimizers
classical simulator
IBM Nighthawk device
Quantinuum Helios device
...

Quantum error-correcting codes (QECC)



Verifying QECCs

QECV: A Verification Framework for Quantum Error Correction Codes

Authors: Anbang Wu, Gushu Li, Hezi Zhang, Gian Giacomo Guerreschi, Yuan Xie, Yufei Ding

Symbolic Execution for Quantum Error Correction Programs

Authors: Wang Fang, Mingsheng Ying | Authors Info & Claims

Proceedings of the ACM on Programming Languages, Volume 8, Issue PLDI • Article No.: 189, Pages 1040 - 1065
https://doi.org/10.1145/3656419

Efficient Formal Verification of Quantum Error Correcting Programs

Authors: Qifan Huang, Li Zhou, Wang Fang, Mengyu Zhao, Mingsheng Ying | Authors Info & Claims

Proceedings of the ACM on Programming Languages, Volume 9, Issue PLDI • Article No.: 190, Pages 1068 - 1093
https://doi.org/10.1145/3729293

Stabilizer Formalism

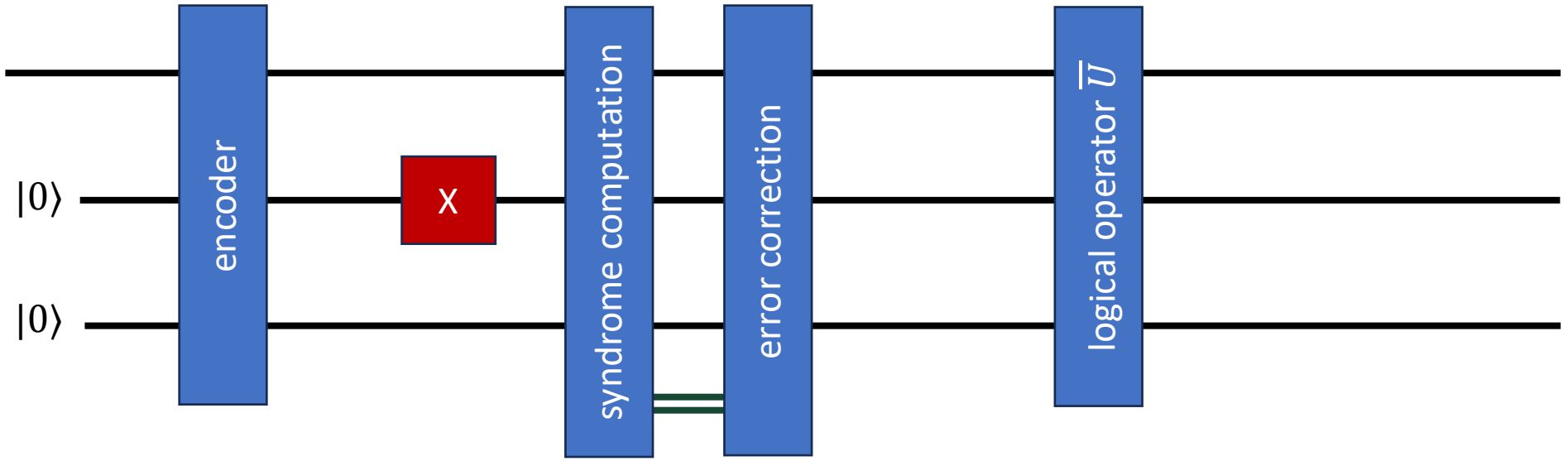
Qubits	Semantics
pure state $\|\varphi\rangle \in \mathbb{C}^{2^n}$	$U \in \mathbb{C}^{2^n \times 2^n}$
mixed state *(probability distribution over pure states)* $\rho = \sum_i p_i \|\varphi_i\rangle\langle\varphi_i\| \in \mathbb{C}^{2^n \times 2^n}$	completely positive maps (CPMs) $\rho \mapsto \rho'$ i.e. $\mathrm{tr}(\rho) \geq 0 \rightarrow \mathrm{tr}(\rho') \geq 0$
stabilizer states *(set of Pauli operators that stabilize the state)* $S \subseteq \mathcal{P}_n \cong \mathbb{Z}_4 \times \mathbb{Z}_2^{2n}$ $\Rightarrow$ $\{ \|\varphi\rangle \mid \forall P \in S, P\|\varphi\rangle = \|\varphi\rangle \}$	Clifford operators as Pauli tableaux: $T \in (\mathcal{P}_n)^{2n}$

efficiently simulatable on a classical computer

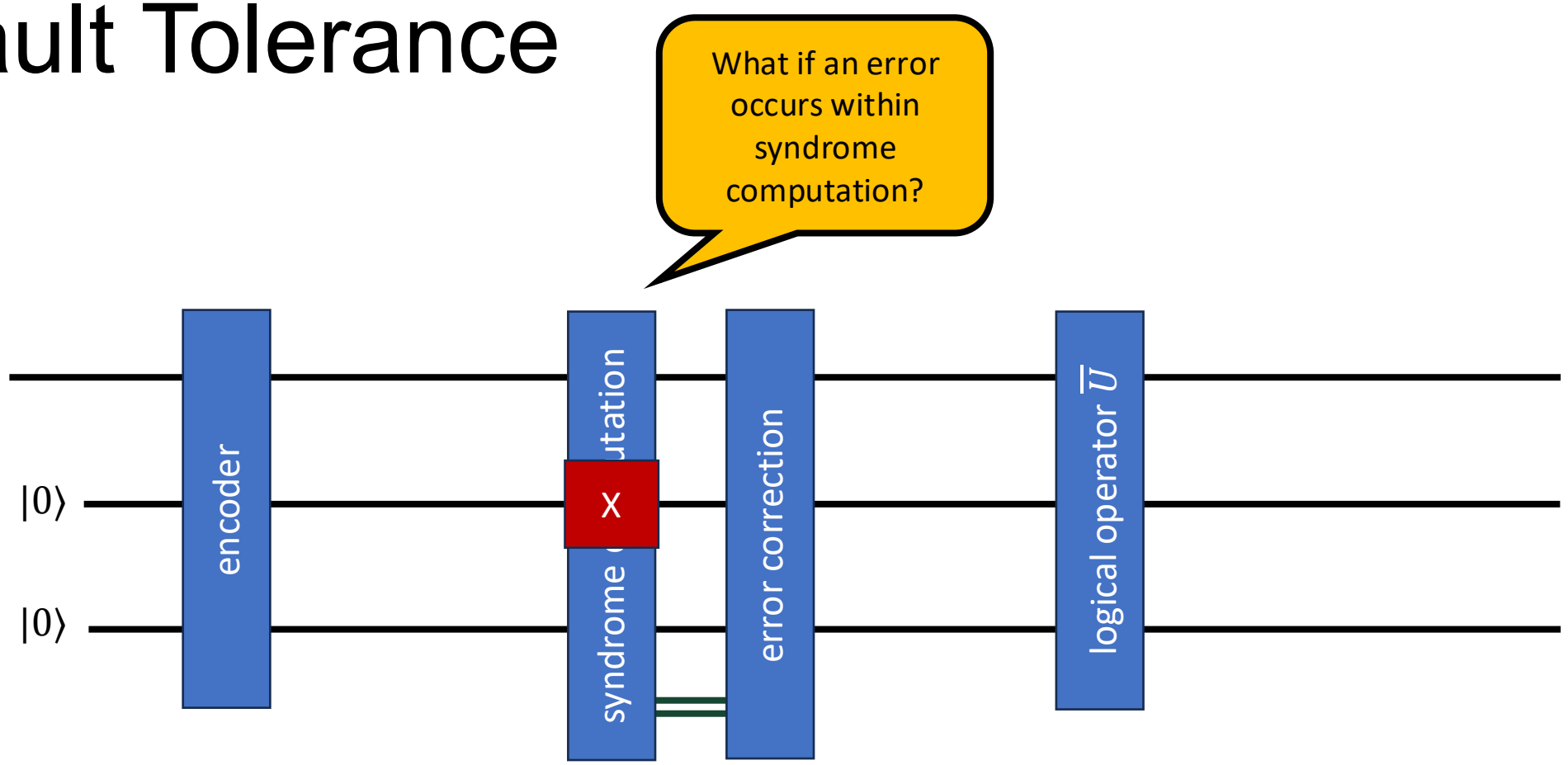
bitvector representation

Daniel Gottesman.
Surviving as a Quantum Computer in a Classical World.
2025.

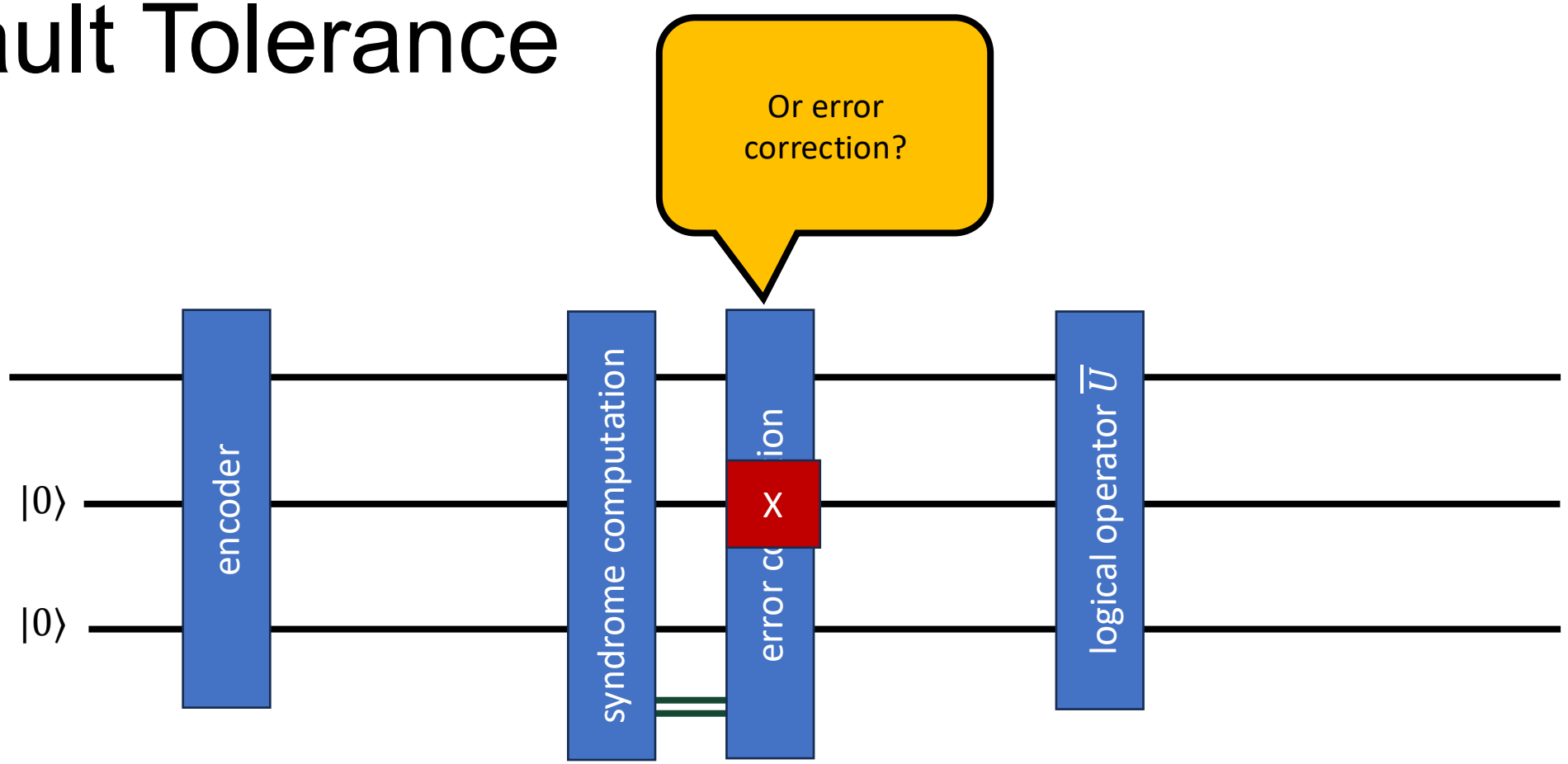
Fault Tolerance



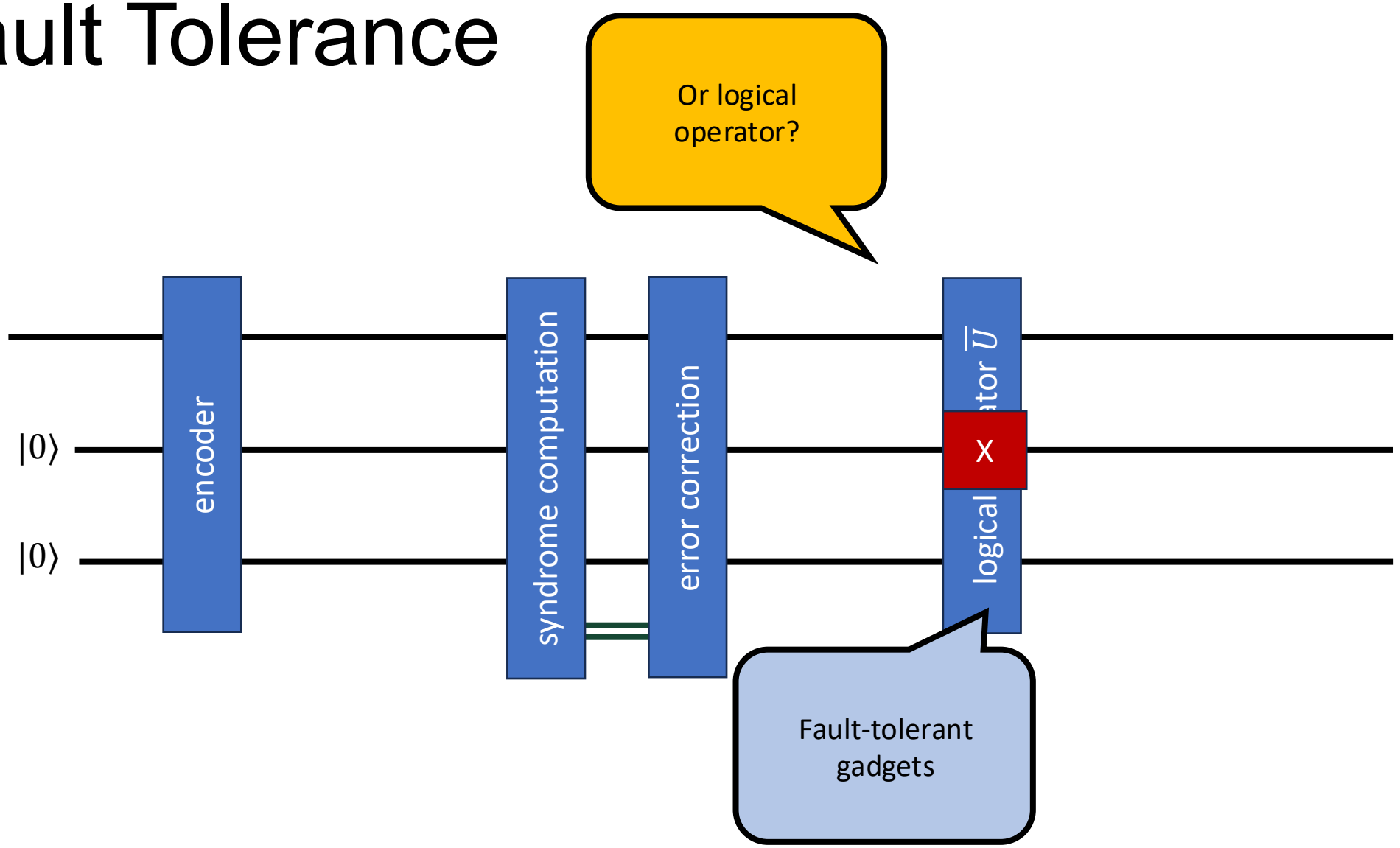
Fault Tolerance



Fault Tolerance



Fault Tolerance



Verifying fault-tolerance of gadgets

- Formalize fault-tolerance
- Symbolic execution with added quantum faults
- Special symbolic execution rules for repeat-until-success
- Plus: magic state distillation
  - Decompose rules into parts to be checked with above techniques

Verifying Fault-Tolerance of Quantum Error Correction Codes

Kean Chen[1(✉)], Yuhao Liu[1], Wang Fang[2], Jennifer Paykin[3], Xin-Chuan Wu[4], Albert Schmitz[3], Steve Zdancewic[1(✉)], and Gushu Li[1(✉)]

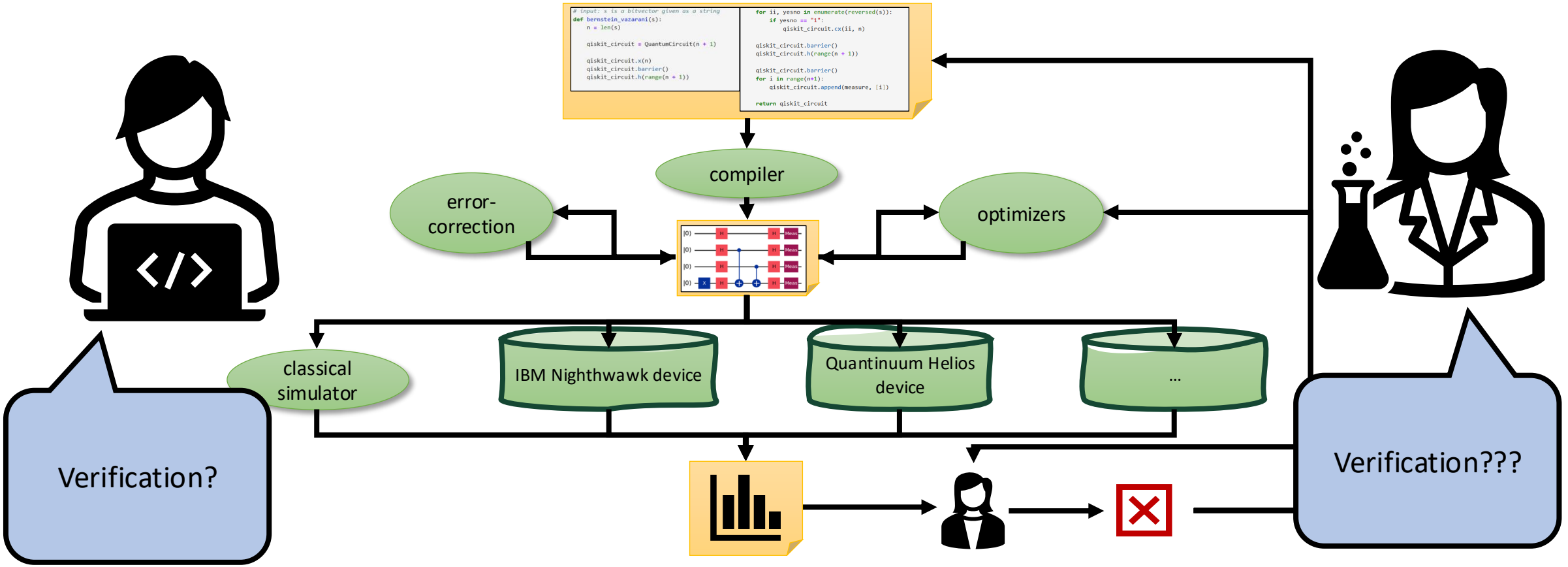
[1] University of Pennsylvania, Philadelphia, USA
{keanchen,stevez,gushuli}@seas.upenn.edu
[2] University of Edinburgh, Edinburgh, UK
[3] Intel Corporation, Hillsboro, USA
[4] Intel Corporation, Santa Clara, USA

Open problems

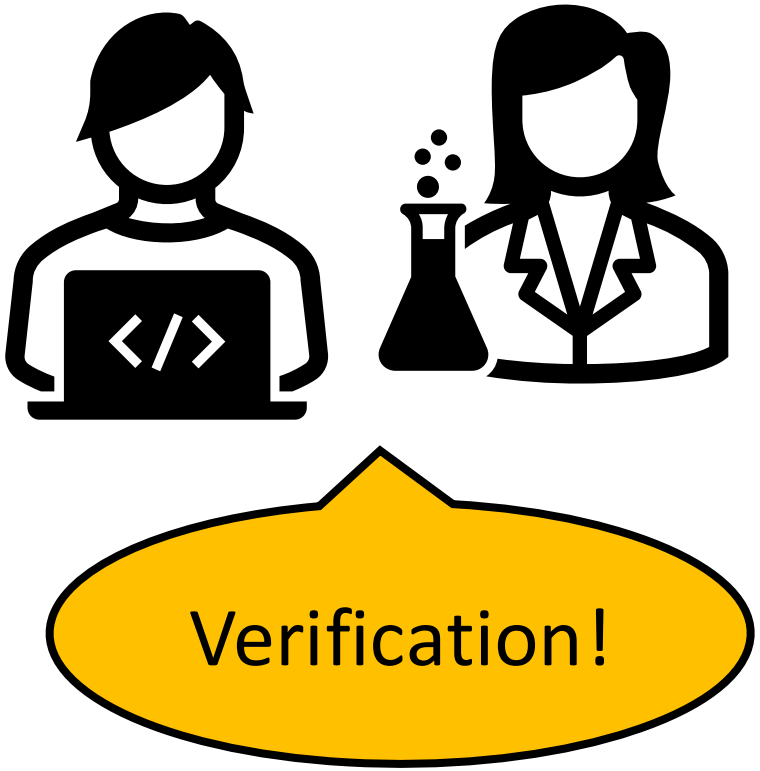
1. Structures to efficiently represent quantum semantics
2. Verifying more than circuits: quantum programs
3. Automation/expressive power tradeoff
4. How to intuitively express quantum specifications?
5. Verification in the presence of errors

6. Encourage adoption from all

Open problem #6: Adoption and collaboration



Open problem #6: Adoption and collaboration



Specification, Semantics, and Verification of Quantum Programs

Questions?

Jennifer Paykin
University of Vermont
jpaykin@uvm.edu

Hiring PhD students!

University of Vermont

Certified Programs and Proofs (CPP)
Jan 13, 2026